

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Triinu Liis Kelder**  
**Juhtmevaba iselaadiva Arduino roboti ehitamine**  
Bakalaureusetöö (9 EAP)

Juhendajad: Anne Villems  
Taavi Duvin  
Alo Peets

TARTU 2016

# **Juhtmevaba iselaadiva Arduino roboti ehitamine**

## **Lühikokkuvõte:**

Käesoleva bakalaureusetöö eesmärgiks on kirjeldada, kuidas ehitada Arduino robotit, mis tuvastab aku tühjenemise ning liigub laadimispunkti, kus robot hakkab laadima juhtmevabalt. Töö esimeses peatükis tutvustatakse Arduino platvormi. Teises peatükis kirjeldatakse aku tühjenemise tuvastamist. Kolmandas peatükis antakse ülevaade juhtmevabast laadimisest. Neljandas peatükis selgitatakse, kuidas jõuab robot laadimispunkti. Viiendas peatükis antakse ülevaade töö tulemustest.

## **Võtmesõnad:**

Juhtmevaba laadimine, Arduino, robotika

**CERCS:** P170 (Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine)

# **Building wireless self-charging Arduino robot**

## **Abstract:**

The main purpose of this bachelor thesis is to describe how to build an Arduino robot which detects low battery and moves to the charging station, where the robot starts to charge wirelessly. The first chapter of the thesis introduces Arduino platform. The second chapter describes detecting low battery. The third chapter gives an overview of wireless charging. The fourth chapter explains how the robot reaches the charging station. The fifth chapter gives an overview of the results of the thesis.

## **Keywords:**

Wireless charging, Arduino, robotics

**CERCS:** P170 (Computer science, numerical analysis, systems, control)

# Sisukord

Sissejuhatus.....	5
1. Arduino ülevaade.....	6
1.1 Arduino tutvustus.....	6
1.2 Arduino arendusplaatide võrdlus .....	9
1.3 Arduino laiendusplaadid .....	10
1.4 Arduino võrdlus Raspberry Pi-ga .....	11
1.5 Arduino programmeerimine.....	13
2. Aku tühjenemise tuvastamine .....	17
2.1 Aku tööpõhimõte .....	17
2.2 Aku laetuse tuvastamine .....	18
2.3 Aku pinge leidmine Arduinol .....	18
2.4 Laadima minemise hetke tuvastamine .....	21
3. Juhtmevaba laadimine.....	23
3.1 Juhtmevaba laadimise ajalugu .....	23
3.2 Juhtmevaba laadimise tööpõhimõte.....	24
3.3 Magnetiline induktsioon .....	26
3.4 Magnetiline resonants .....	27
3.5 Qi juhtmevaba laadimise standard .....	27
3.6 Aku laadimine juhtmevabalt .....	28
4. Laadimispunkti liikumine .....	30
4.1 Arduino ühendamine mootoritega .....	30
4.2 Mootorite liigutamine .....	33
4.3 Laadimispunkti asukoha tuvastamine infrapuna sensorite abil .....	35

4.3.1 Infrapunakiirgus .....	36
4.3.2 Infrapuna sensorite ühendamine Arduinoga .....	36
4.3.3 Infrapuna sensorite abil laadimispunkti leidmine .....	38
4.4 Ultraheli sensor .....	41
4.5 Laadimispunkti liikumise algoritm .....	44
5. Tulemused .....	46
5.1 Roboti ehitamine .....	46
5.2 Roboti laadimispunkti jõudmine .....	47
Kokkuvõte .....	49
Kasutatud kirjandus .....	50
Lisad .....	58
I. Roboti kood .....	58
II. Litsents .....	64

## Sissejuhatus

Juhtmevaba laadimise kättesaadavus on viimastel aastatel kasvanud. Üha rohkem nutitelefone on võimalik laadida juhtmevabalt. See tähendab, et enam pole vaja seadme otsest kontakti laadijaga ning laadija saab vajadusel kasvõi mõne mööblieseme sisse peita. Peale nutitelefoni saab juhtmevaba laadimist kasutada ka teiste seadmete laadimiseks.

Uuematel nutitelefonidel on juhtmevaba laadimise vastuvõtja sisse ehitatud. Teiste seadmete jaoks pakutakse turul universaalset vastuvõtjat, mille saab ühendada seadme Micro-USB ühenduspesasse. Käesolevas töös laetakse juhtmevabalt akupanka, mis on ühendatud Arduino mikrokontrolliga.

Arduinot kasutatakse käesolevas töös, et ehitada robot, mis sõidab vabalt toas ringi. Mingi hetk saab aga roboti akupank tühjaks ning roboti edasi liikumise võimaldamiseks on seda vaja laadida. Tõukas on jälgida, millal aku tühjaks saab ning seejärel see laadima panna. Selle asemel on mugavam, kui robot saaks ise aru, et aku on tühjenemas ning läheb ise laadimispunkti, kus end juhtmevabalt laadida.

Turul on juba robottolmuimejad ning -muruniidukid, mis liiguvad ise laadimispunkti, kuid enamasti ei kasuta laadimiseks juhtmevaba laadimist. Praegu on siiski vajalik kontakt laadijaga. See tähendab, et seade peab olema õigesti laadijaga ühenduses, et laadima hakata.

Antud bakalaureusetöö eesmärk on luua eestikeelne juhend, kuidas ehitada robotit, mis liigub aku tühjenemisel laadimispunkti, kus juhtmevabalt laadida. Arduino mikrokontroller ning laadimispunkti jõudmiseks vajalikud sensorid on Eestis kättesaadavad ning taskukohase hinnaga. Juhtmevaba laadijat ja –vastuvõtjat saab osta internetipoodidest.

Käesolev töö koosneb viiest peatükist. Töö esimeses peatükis antakse ülevaade Arduino platvormist. Teises peatükis kirjeldatakse, kuidas tuvastada aku tühjenemist robotil. Kolmandas peatükis tutvustatakse juhtmevaba laadimist. Neljandas peatükis selgitatakse, kuidas robot tuvastab laadimispunkti asukoha. Viiendas peatükis antakse ülevaade töö tulemustest.

# 1. Arduino ülevaade

Käesolevas peatükis antakse ülevaade Arduino platvormist. Tutvustatakse Arduino programmeerimist, selle arendus- ja laiendusplaate ning võrreldakse Arduino ning Raspberry Pi platvormi.

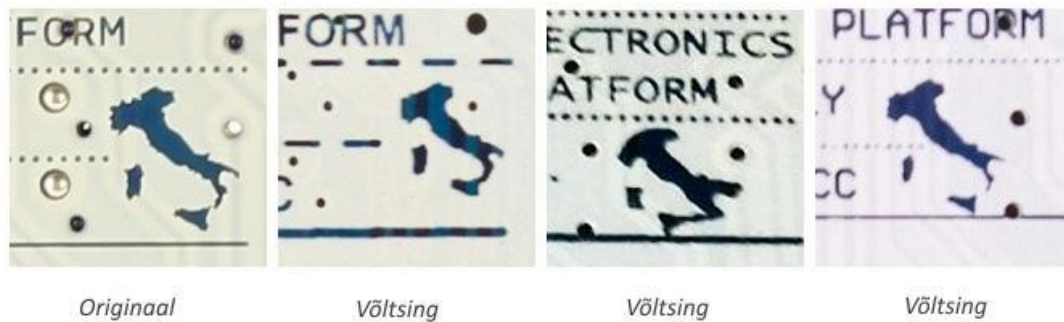
## 1.1 Arduino tutvustus

Arduino on avatud lähtekoodi ja -disainiga mikrokontrollerplatvorm elektroonika prototüüpimiseks. Arduinole saab külge ühendada erinevaid andureid (termo-, valgus-, vibratsiooni-, heli- jt andurid). Andurite abil saab Arduino sisendi ning edasi reageerib mikrokontroller vastavalt koodile ning käivitab mõne väljundseadme (mootor, valgusdiod, ekraan jne) [1].

Arduinoga programmeerimiseks on algajatel kõige lihtsam kasutada Arduino integreeritud arenduskeskkonda (IDE - *Integrated development environment*) ning Arduino programmeerimiskeelt, mis tugineb *Wiring* keskkonnale [2]. Arduino programmeerimiskeel baseerub C/C++ programmeerimiskeelele. Kompileerimisel tehakse koodis väikesed muudatused ning antakse edasi C/C++ (täpsemalt avr-g++) kompilaatorile. Erinevalt C-programmeerimiskeeles kirjutatud programmist tuleb töötava programmi jaoks kasutajal defineerida kaks funktsiooni: *setup()*, mis kutsutakse välja üks kord programmi alguses ja millega seadistatakse parameetrite algväärtused, ning *loop()*, mis kutsutakse korduvalt välja ja mis võimaldab oludel muutuda ja programmil vastavalt reageerida [3,4,5,6].

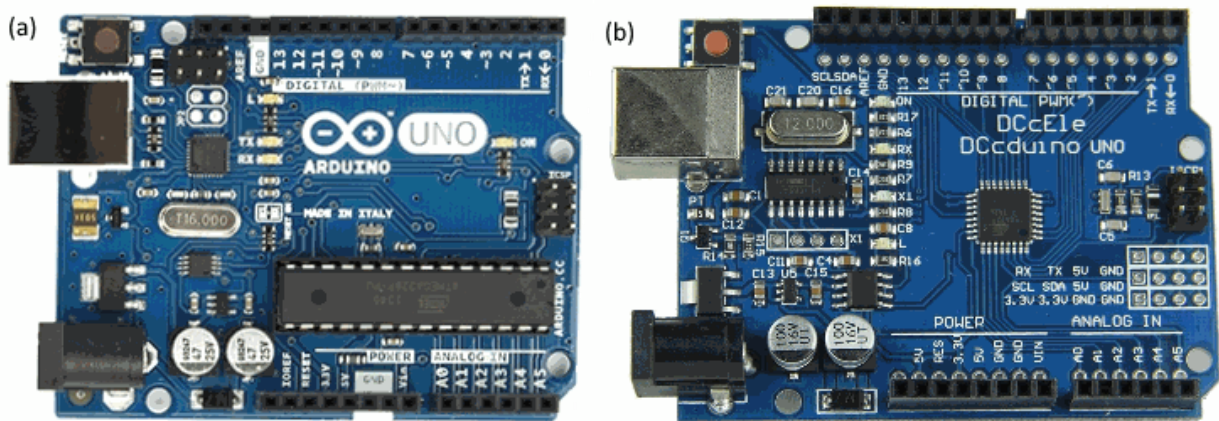
Arduino kaubamärgi all müüakse tooteid ainult Ameerika Ühendriikides. Teistes riikides müüakse samu tooteid Genuino nime all [7]. Genuino nime all müüakse tooteid teistes riikides, kuna üks asutajatest Gianluca Martino registreeris Arduino nime Itaalias teiste asutajate teadmata. Jõudmata omavahel kokkuleppele Arduino kaubamärgi kasutamise üle Itaalias, kus Arduino on väga populaarne, lõpetati koostöö Gianlucaga ning loodi uus kaubamärk Genuino, et saaks ka väljaspool Ameerika Ühendriike tooteid müüa [8,9]. Lisaks Arduinole ja Genuinole esineb ka suur hulk võltsinguid, derivaate ja kloone. Kuna Arduino on avatud lähtedisainiga, on lubatud toota sarnase disaini ja võimalustega tooteid, kuid mitte Arduino kaubamärgi all. Võltsingud müüvad tihti ennast Arduino nime all ning kasutavad Arduino logo, kuid kvaliteet on tavaliselt madalam (vt joonis 1). Võltsingud on keelatud ning

lähevad vastuollu autoriõigustega. Arduino tootja üritab neid tuvastada, et inimestele ei müüdaks praaktooteid Arduino nime all [10].



**Joonis 1.** Märk arendusplaadi tagaküljel Arduino originaaltootel ja võltsingutel [10]

Derivaadid põhinevad Arduino toodetel, kuid neil on tihtipeale natuke erinev paigutus ja tunnused. Need tooted aitavad koguda Arduino toodetel populaarsust. Kloonid on tavaliselt väga sarnased päris Arduino toodetega, kuid müüvad end teise nime all (tavaliselt sisaldub nimes "Ardu" või "duino") [10]. Käesolevas töös kasutatakse Arduino kloni DCcEle DccDuino UNO plaati (vt joonis 2 (b)). DCcEle DccDuino UNO on peaaegu kaks korda odavam kui Arduino UNO [11,12]

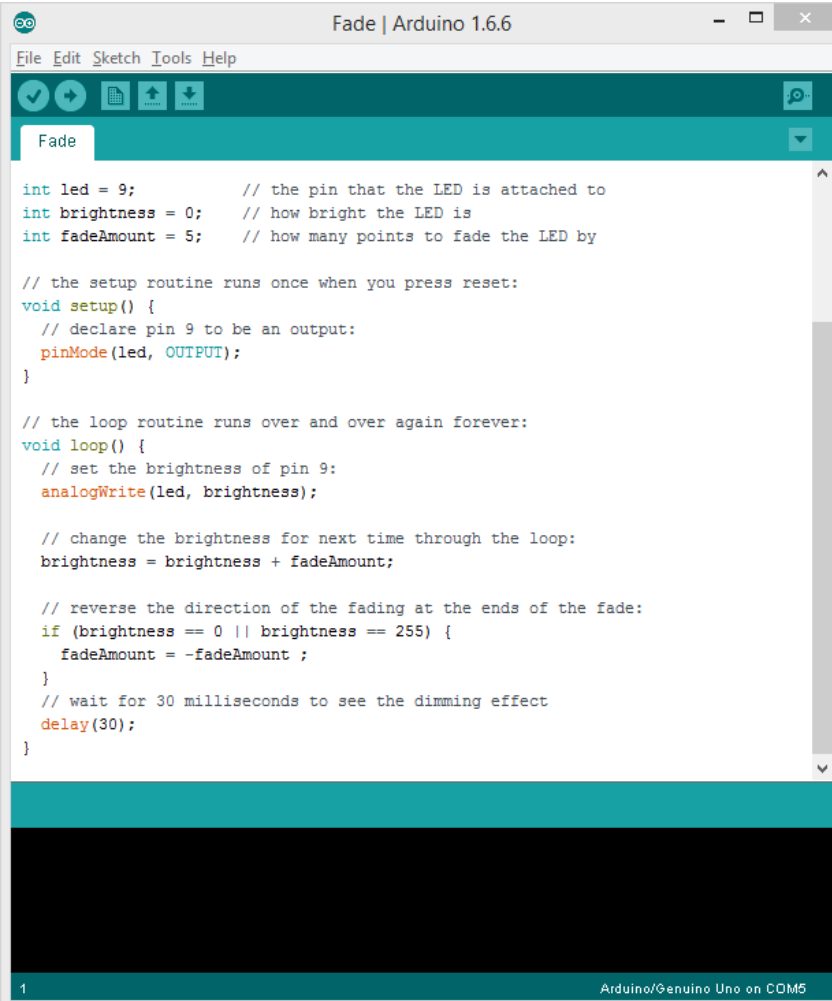


**Joonis 2.** Arduino UNO (a) ja DCcEle DCcduino UNO (b) [12,13]

Käesoleva töö teoreetiline osa põhineb Arduino originaaltootel, kuna on kõige populaarsem ning teiste versioonide aluseks olev tehnoloogia. Enamik analüüsi ja õpetusi on kehtivad ka kloonide ja derivaatide korral.

Arduino on sobiv platvorm nii algajatele kui ka kogenenud kasutajatele. Arduino IDE-ga on kaasas palju näidisprogramme, et alustamine oleks võimalikult lihtne. Samas on Arduino programmeerimiskeel paindlik, kuna sellega saab kogenenud kasutaja teha ka keerulisemaid projekte, kombineerides või muutes erinevaid näiteprogramme ning lisades väliseid teeke.

Arduino IDE-t saab tasuta alla laadida Arduino kodulehelt. Programm on sobiv ka enamiku kloonidega. Tihti tuleb kloonide kasutamiseks installeerida vajalikud draiverid, et saaks Arduino IDE-t nendega kasutada. Arduino IDE käivitamisel avatakse uus fail, milles on defineeritud funktsioonid *setup()* ja *loop()*. Järgmiseks saab avada mõne näiteprogrammi või kirjutada ise koodi juurde ning seejärel see Arduino arendusplaadile üles laadida (vt joonis 3).



```
Arduino IDE 1.6.6 - Fade

File Edit Sketch Tools Help

Fade

int led = 9;           // the pin that the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

1 Arduino/Genuino Uno on COM5

**Joonis 3.** Ekraanitõmmis Arduino integreeritud arenduskeskkonnast



Kõige levinum arendusplaat on Arduino UNO, kuid peale selle on veel teisi arendusplaate, mis sobivad erinevateks projektideks. Levinumad neist tuuakse välja järgmises punktis.

## 1.2 Arduino arendusplaatide võrdlus

Arduinol on palju erinevaid arendusplaate. Mõned levinumad tooted on Arduino UNO, Arduino 101, Arduino Pro, Arduino Mega2560 ning Arduino Zero [14] (Tabel 1). Neist kõige levinum on Arduino UNO ning selle jaoks on saadaval palju õpetusi sensorite ühendamiseks ning programmeerimiseks. Arduino stardikomplekt, mis on mõeldud Arduinoga alustamiseks, sisaldab lisaks erinevatele sensoritele ja juhtmetele samuti Arduino UNO-t. On ka teisi arendusplaate, mis sobivad teatud projektideks paremini. Nendest levinumad tuuakse välja järgmises tabelis.

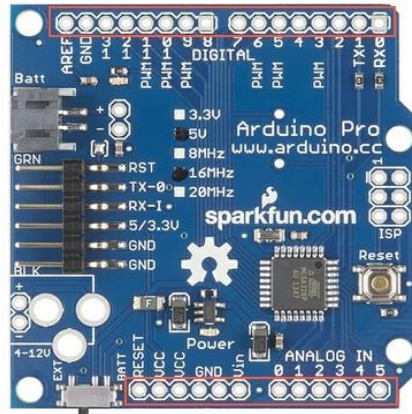
**Tabel 1.** Arduino plaatide võrdlus [15]

Arduino	UNO	101	Pro	Mega 2560	Zero
Mikrokontroller (Taktsagedus)	ATmega328P (16 MHz)	Intel Curie (32 MHz)	ATmega328 (16 MHz)	ATmega2560 (16 MHz)	ATSAMD21G18 (48 MHz)
Tööpinge / Sisendpinge	5V / 7-12V	3.3V / 7-12V	5V / 5-12V	5V / 7-12V	3.3V / 7-12V
Analoogpesade arv Sisse/Välja	6 / 0	6 / 0	6 / 0	16 / 0	6 / 1
Digitaalpesade arv IO/PWM	14 / 6	14 / 4	14 / 6	54 / 15	14 / 10
USB	Tavaline	Tavaline	-	Tavaline	2 Micro
Hind	\$24.95 [11]	\$30.00 [16]	\$14.95 [17]	\$39.74 [18]	\$48.71 [19]

Arduino UNO on alustamiseks kõige parem, kuna sellel on kõik vajalik olemas ning leidub palju õpetusi programmeerimiseks ja sensorite ühendamiseks [20]. Stardikomplektiga tulevad lisaks Arduino UNO-le kaasa veel mõned andurid ning vajalikud juhtmed nende ühendamiseks, et saaks kohe mõne projekti teha [21].

Arduino 101-l on samad omadused, mis Arduino UNO-l. Nendele lisaks on 101-l *Bluetooth*'i tugi ning 6-teljeline kiirendusmõõtur/güroskoop [22].

Arduino Pro on mõeldud püsivaks paigalduseks. Sellel ei ole eelnevalt paigaldatud ühendusklemme (ingl k *headers*) (vt joonis 4), seega tuleb kasutada erinevaid ühenduspesasid või joota juhtmed otse plaadi külge [23].



**Joonis 4.** Arduino Pro ühendusklemmideta (joonisel punaselt) [23]

Arduino Mega 2560 on sarnane Arduino UNO-ga. Kõige suurem erinevus on digitaalpesade arv, mis UNO-l on 14 ning Mega 2560-l 54. Arduino Mega 2560-le saab külge ühendada enamikke UNO laiendusplaate [24].

Arduino Zero on Arduino UNO edasiarendus. Sellel on ARM protsessor, mis on võimsam ning on lisatud Atmeli sisseehitatud silur, mis teeb tarkvara silumise mugavamaks ning toetab virtuaalset COM porti, mida saab kasutada seadme ja alglaaduri programmeerimiseks [25].

Targemate ja paremate süsteemide loomiseks saab kasutada laiendusplaate ja sensoreid, millega tutvume järgmises punktis.

### 1.3 Arduino laiendusplaadid

Rohkemate võimaluste jaoks saab kasutada Arduino laiendusplaate, mis laiendavad Arduino arendusplaadi võimalusi. Laiendusplaadid kinnitatakse arendusplaadi peale. Ilma arenduspladita laiendusplaate kasutada ei saa [26,27].

Laiendusplaadid annavad võimaluse teha Arduinoga veelgi rohkem. Näiteks saab arendusplaadi peale kinnitada LCD-tähtekraaniga laiendusplaadi, millel saab kuvada teksti. Käesolevas töös kasutatakse prototüüpimise laiendusplaati. Seda on vaja selleks, et saaks

arendusplaadiga ühendada rohkem andureid. Prototüüpimise laiendusplaadile saab vajalikud osad külge joota või lisada lahtise maketeerimislauda, millega saab tekitada vastavalt vajadusele erinevaid vooluringe (vt joonis 5) [28]. Antud töös kasutatakse lahtist maketeerimislauda, mis võimaldab vajadusel juhtmeid ümber tõsta.

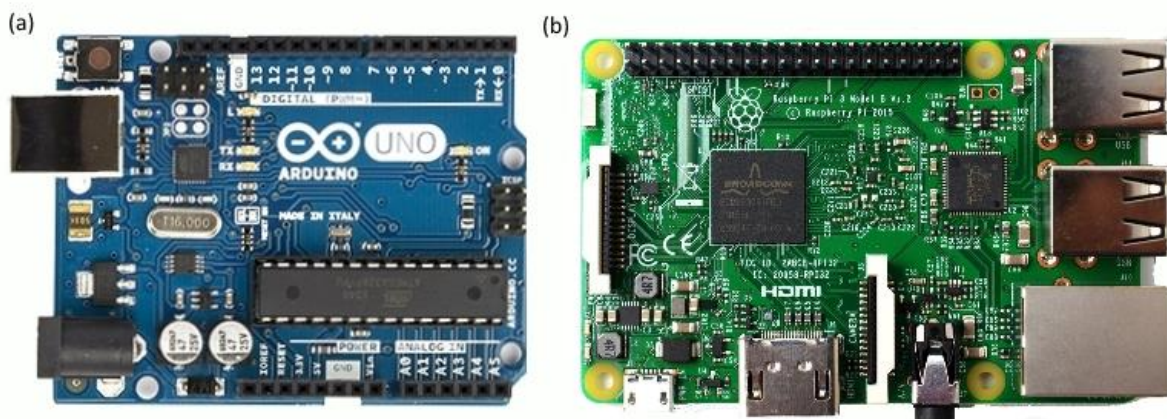


**Joonis 5.** Arduino prototüüpimise laiendusplaat koos maketeerimislauga (valge aukudega plaat joonisel) [29]

Peale Arduino arendusplaadi on veel sarnaseid platvorme. Üks neist on Raspberry Pi, millega tutvume järgmises punktis.

## 1.4 Arduino võrdlus Raspberry Pi-ga

Kõige tuntum Arduino UNO-ga sarnane platvorm on Raspberry Pi. Pealtnäha näevad platvormid sarnased välja (vt joonis 6) ning on ka samas hinnaklassis, kuid funktsionaalsus on erinev ning teatud projektideks sobib kindel platvorm [30].



**Joonis 6.** Arduino UNO (a) ja Raspberry Pi 3 (b) [31,32]

Põhiline erinevus Arduino ja Raspberry Pi vahel on nende funktsionaalsus. Arduino mikrokontroller sobib rohkem riistvaralähedaste projektide jaoks. Arduinole saab külge ühendada andureid ning seejärel programmeerida tarkvara, mille abil neid kasutada. Andurite abil saab Arduino informatsiooni väliskeskkonnast ning reageerib sellele vastavalt [30].

Raspberry Pi on mini-arvuti, kus jookseb Linuxi operatsioonisüsteem. Raspberry Pi-le saab külge ühendada hiire ja klaviatuuri ning kasutada seda internetis veebilehitsemiseks või mängimiseks. Raspberry Pi-d kasutatakse tavaliselt tarkvara projektide jaoks. Ka Raspberry Pi-le saab ühendada külge sensoreid, et teha veelgi rohkem projekte [30].

Kõige suurem puudus Raspberry Pi-l võrreldes Arduinoga on ADC ehk analoog-digitaalmuunduri (ingl k *analog-to-digital-converter*) puudumine. See tähendab, et analoogsisendit Raspberry Pi-l ei ole ning seega ei saa selle abil pinget või voolu mõõta. Analoogsisendit kasutavad paljud andurid (temperatuuri-, valgusandur) [33].

Raspberry Pi programmeerimiseks võib kasutada erinevaid keeli. Kõige enam kasutatakse Pythonit, kuna seda on algajatel lihtne õppida [34]. Arduino IDE-s kasutatakse Arduino programmeerimiseks C-programmeerimiskeelt [3].

**Tabel 2.** Arduino UNO ja Raspberry Pi 3 võrdlus [30,35,36]

	Arduino UNO	Raspberry Pi 3
Operatsioonisüsteem	Arduino alglaadur	<i>Linux</i>
Milleks mõeldud	Riistvaralähedased projektid	Tarkvara projektid
I/O pesade arv	14 digitaalpesa, 6 analoogpesa	26 GPIO-pesa ( <i>General-purpose input/output</i> ) [37]
Taktsagedus	16 MHz	1.2 GHz
Audio/video väljund	Põhiline funktsionaalsus võimalik programmeerida	Jah
Internet	Laiendusplaadi abil	Jah
Multitöötlus (ingl k <i>multitasking</i> )	Ei	Jah
Hind	\$21.99 [13]	\$41.79 [38]

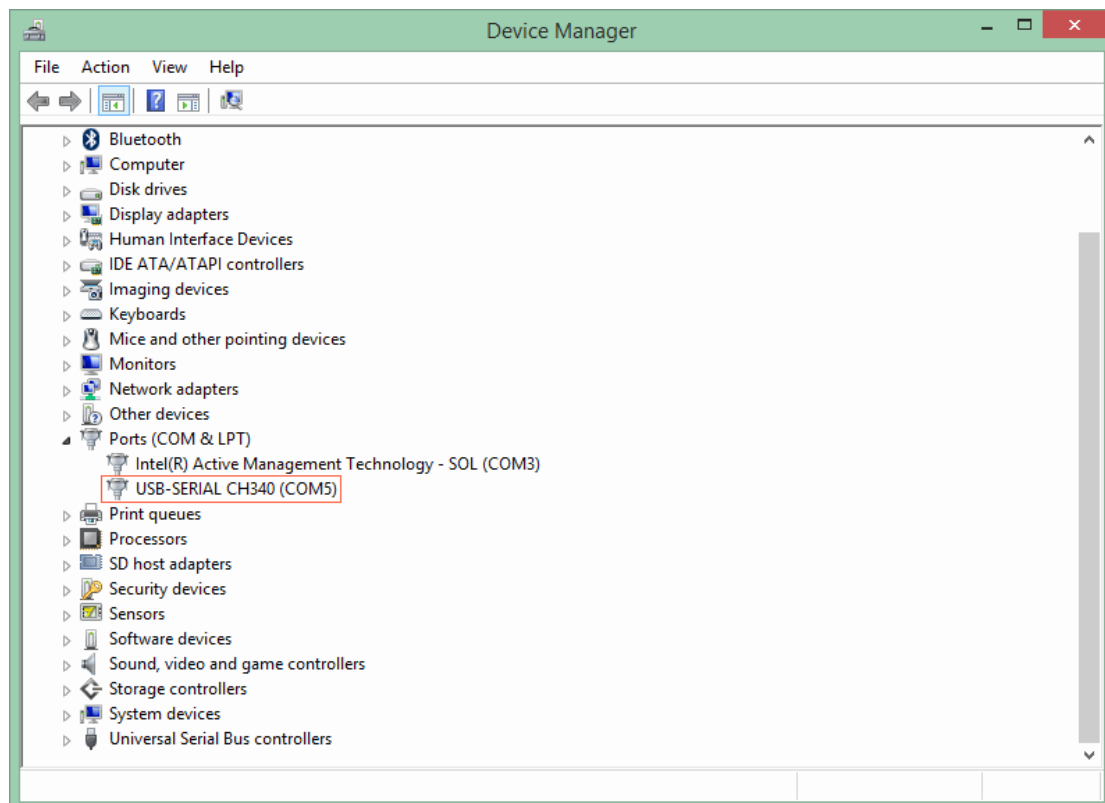
Tabelist 2 on näha, et Raspberry Pi on kiirema taktsagedusega ning võimaldab multitöötlust, mida Arduino ei võimalda. See on vajalik, et jooksutada *Linux*'i operatsioonisüsteemi. Raspberry Pi 3 maksab ka peaaegu 2 korda rohkem kui Arduino UNO.

Kuna Arduino puhul on olulisel kohal programmeerimine, tutvustatakse seda lähemalt järgmises punktis.

## 1.5 Arduino programmeerimine

Antud töös kasutatakse Arduino programmeerimiseks Arduino IDE-t, mis on loodud just Arduino ja Genuino programmeerimiseks. Keskkonda saab kasutada ka Arduino kloonide ja derivaatide programmeerimise jaoks, kuna nende riistvara on analoogne, kuid enamasti tuleb installeerida eraldi vajalikud draiverid.

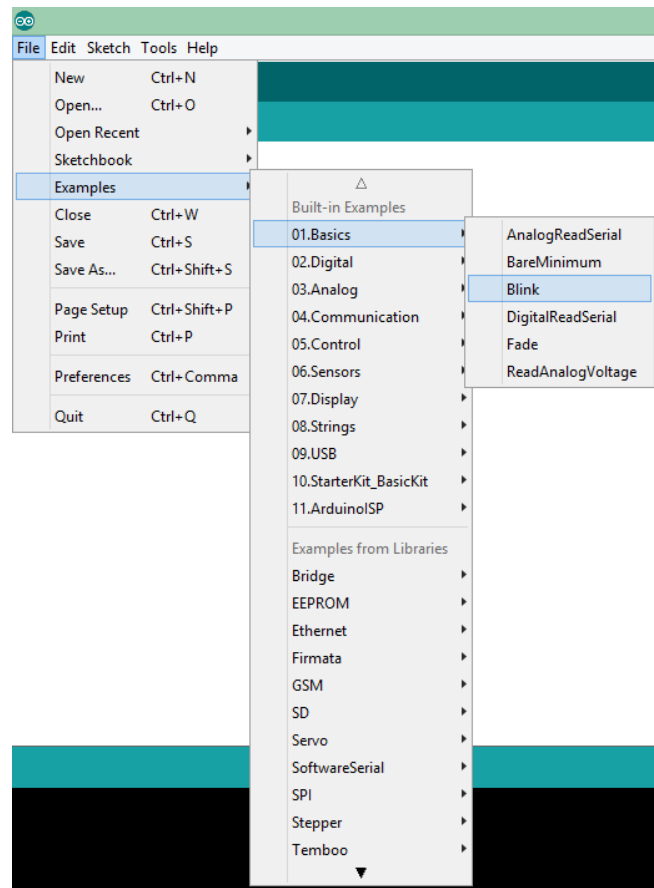
Selleks, et Arduino plaadile programm üles laadida, tuleb arendusplaat ühendada arvutiga USB-kaabli abil ning installeerida plaadi draiverid, et arvuti plaadi ära tunneks. Seejärel on vaja tuvastada, millisesse porti Arduino ühendatud on. Operatsioonisüsteemis *Windows* tuleb selleks avada programm *Device Manager*, mille leiab kirjutades stardimenüü otsingusse „device“. Programmi avades tuleb valida *Ports (COM & LPT)*. Edasi tuleb leida avanenud nimekirjast ühendatud Arduino arendusplaat. Kui nime järgi ole võimalik leida, ühendada USB lahti ning seejärel see taasühendada ning jälgida, milline nimi tekkis juurde. Nime taga on pordi number (COMx, kus x on pordi number) (vt joonis 7). Arduino IDE avamisel saab nüüd valida sobiva pordi, valides menüüst *Tools*, seejärel *Port*, viimaks valida õige port.



**Joonis 7.** Ekraanitõmmis õige pordi leidmisest programmi *Devices Manager* abil

Alustamiseks on Arduino IDE-s sisse kirjutatud hulk näidisprogramme. Nende avamiseks tuleb valida menüüst *File*, seejärel *Examples* ning avanenud menüüst sobiv näidisprogramm. Alustamiseks sobib kõige paremini alamenüü *Basics* ning programm *Blink* (vt joonis 8), mis paneb Arduino arendusplaadil oleva LED-lambi põlema ning sekundi pärast kustutab selle ära. Alustamiseks on see programm hea, kuna on kergesti arusaadav ning kasutaja saab kohe näha LED-lambi vilkumise järgi, kas Arduino on õigesti ühendatud.

Vahel tuleb kasutada programmi kirjutamisel teeki, mida Arduino IDE-s olemas pole. Koostatud on palju õpetusi, milles on kasutatud isetehtud teeki. Teekide kood pannakse tavaliselt üles versioonihaldussüsteemi *Github* või pannakse allalaadimise link õpetuse lehele. Pärast teegi allalaadimist tuleb selle kasutamiseks valida menüüst *Sketch*, seejärel *Include Library* ning tekkinud menüüst valida *Add .ZIP Library*. Ilmunud aknas tuleb liikuda kausta, kus allalaaditud teek asub, ning see valida. Nüüd saab kasutada uut teeki ning selle näidisprogramme.



**Joonis 8.** Näiteprogrammi *Blink* avamine Arduino IDE-s

Enne kohustuslikke meetodeid *setup()* ja *loop()* kirjeldatakse kasutatud välised teegid reaga `#include <Teegi_nimi.h>` ning luuakse objektid, mida hiljem programmis kasutama hakatakse. Tihti määratakse enne meetodeid ära, milliseid pesasid sensorid kasutavad (vt joonis 9). Meetodis *setup()* seadistatakse sensorid nii, et need on kasutamiseks valmis. Meetodis *loop()* kasutatakse sensoreid ning reageeritakse muutustele vastavalt, kuna meetod kutsutakse korduvalt välja. Lisaks nendele meetoditele saab teha veel teisi meetodeid, mida saab mitu korda välja kutsuda, et sama koodi ei peaks mitu korda kirjutama.

```

#include <Servo.h>

Servo myservo; // loo servo objekt, et kontrollida servot
// enamikel plaatidel saab luua kaksteist servo objekti

int pos = 0;    // muutuja, et säilitada servo asendit

void setup() {
  myservo.attach(9); // ühendab servo objekti pesaga 9
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // läheb 0 kraadist 180 kraadini
    // in steps of 1 degree
    myservo.write(pos);           // ütleb servole, et minna asendisse muutujas 'pos'
    delay(15);                    // ootab 15 millisekundit, et servo jõuaks asendisse
  }
  for (pos = 180; pos >= 0; pos -= 1) { // läheb 180 kraadist 0 kraadini
    myservo.write(pos);           // ütleb servole, et minna asendisse muutujas 'pos'
    delay(15);                    // ootab 15 millisekundit, et servo jõuaks asendisse
  }
}

```

### Joonis 9. Näidisprogrammi *Sweep* kood

Arduino platvormi kasutatakse antud töös, et ehitada robot, mis tuvastab aku pinget ning liigub aku tühjenemisel laadimispunkti. Järgmises peatükis kirjeldatakse, kuidas aku tühjenemist tuvastada.



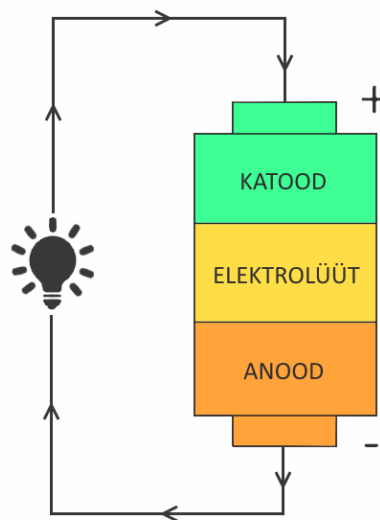
## 2. Aku tühjenemise tuvastamine

Käesolevas peatükis tutvustatakse aku tööpõhimõtet ning selgitatakse, kuidas Arduino tuvastab selle tühjenemist, et hakata õigel ajal liikuma laadimispunkti.

### 2.1 Aku tööpõhimõte

Akupatarei on patarei, mida saab korduvalt laadida ning kasutada. Taaslaetavat akupatareid nimetatakse ka sekundaarelemendiks. Patareid, mida uuesti laadida ei saa, nimetatakse primaarelemendiks [39,40].

Aku koosneb kahest elektroodist: anood, mis on negatiivne pool, ja katood, mis on positiivne pool. Elektrod on elektrijuht, mis ühendab elektriahela osi. Täislaetud akul on elektronid anoodis ning tahavad liikuda katoodi. Anoodi ja katoodi vahel on aga elektrolüüt, mis ei lase elektronidel otse katoodi liikuda. Selleks tuleb anoodi ja katoodi vahele luua suletud vooluring, mille abil elektronid liikuda saaksid. Elektronide liikumise tõttu tekib vool (vt joonis 10) [39,40,41,42].



**Joonis 10.** Patarei tööpõhimõte [42]

Aku laadimise alustamiseks tuleb esmalt tuvastada, et see on tühjenemas. Aku laetuse tuvastamist kirjeldatakse järgmises punktis.

## 2.2 Aku laetuse tuvastamine

Antud töös kasutatakse taaslaetavaid akupatareisid. Tuleb leida viis, kuidas tuvastada akupatareide laetus, et aku tühjenemisel laadimispunkti liikuda.

Akupatareidel hakkab kasutamise käigus pinge langema. Pinge on laetud osakeste potentsiaal teha tööd. Mida suurem on aku pinge, seda suurem on energia vool. See tähendab, et kui aku pinge väheneb, kahaneb aku võime teha tööd ehk aku tühjeneb [43,44].

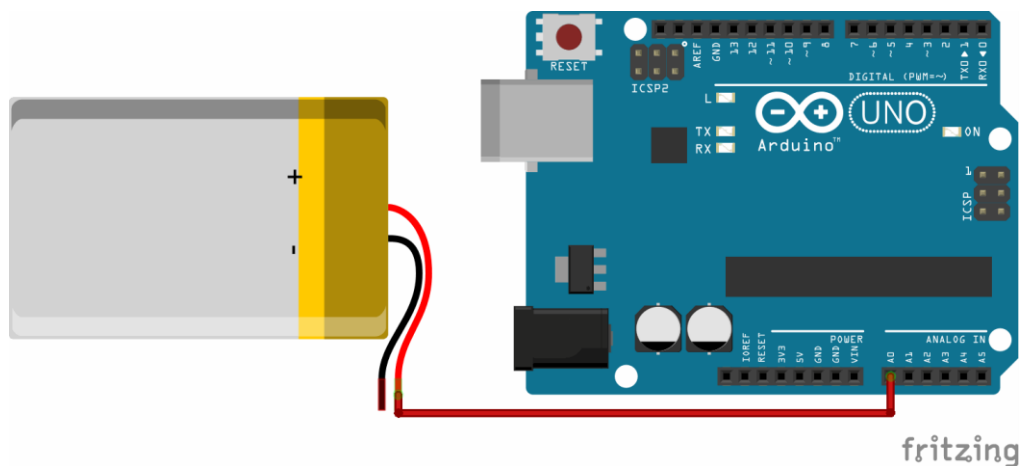
Põhimõtteliselt mida rohkem elektrone on anoodis, seda suurem on aku pinge. Elektronide liikumisel katoodi väheneb nende arv anoodis ning pinge väheneb.

Selleks, et tuvastada hetk, millal laadima hakata, tuleb Arduino arendusplaadil leida aku pinge. Järgmises punktis seletatakse, kuidas leida aku pinget Arduinol.

## 2.3 Aku pinge leidmine Arduinol

Käesolevas töös kasutatakse Li-Po aku tehnoloogial põhinevat 5-voldist akupanka. Akupank ühendatakse Arduino arendusplaadiga USB-kaabli abil.

Lisaks ühendatakse eraldi akupanga positiivne klemm Arduino analoogsisendisse (a0), et lugeda aku pinge väärtust (vt joonis 11)



**Joonis 11.** Aku positiivse klemmi ühendamine Arduino analoogsisendiga

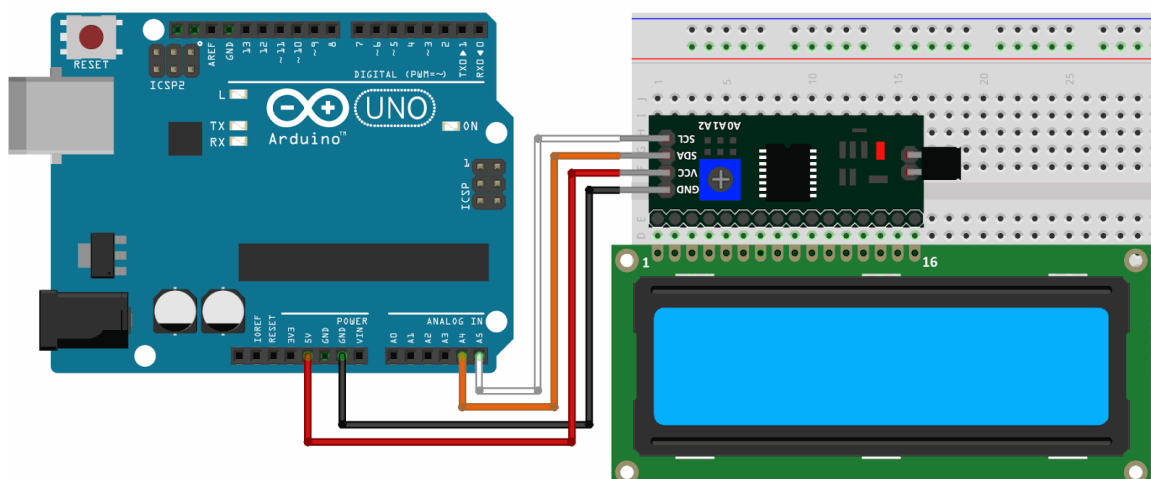
Analoogsisendis teisendatakse pinge 0-5V ümber väärtusteks 0-1023, mis tuleneb 10-bitilisest analoog-digitaalmuundurist [45,46]. Õige pinge leidmiseks tuleb teha joonisel 12 olev arvutus.

$$U = \frac{\textit{sensori väärtus}}{1023} * 5$$

**Joonis 12.** Pinge leidmine analoogsisendist [47]

Kuna Arduino arendusplaat peab olema pinge leidmisel ühendatud akupangaga, ei saa väärtuse kontrollimiseks kasutada jadapordi monitori, kuhu saab teksti väljastada siis, kui arendusplaat on arvutiga ühendatud. Selleks on lisaks võetud LCD-tähtekraan, et sinna pinge väärtus printida.

Pinge leidmisel väljastatakse leitud väärtus selle kontrollimiseks I2C-tüüpi LCD-tähtekraanile. I2C-tüüpi LCD-tähtekraani on kasutatud antud töös põhjusel, et ühendada on vaja vaid 4 juhet. Ilma I2C-kontrollerita LCD-tähtekraanil on vaja aga ühendada 16 juhet ning väikesel maketeerimislaual ei ole neid mugav ühendada. I2C-tüüpi LCD-tähtekraani kasutamiseks tuleb LCD-tähtekraanile külge joota I2C LCD-kontroller või kasutada kontrolleriga ühendamiseks maketeerimislauda. Käesolevas töös on I2C LCD-kontroller LCD-tähtekraanile külge joodetud. Kontroller on Arduinoga ühendatud 4 juhtme abil: toide (punane), baas (must) ja kaks signaali (SDA (oranž) ja SCL (valge)) (vt joonis 13). SDA on andmesignaal ning SCL on taktsignaal [48]. SDA ja SCL on mõeldud ühendamiseks Arduinol vastavatesse pesadesse. Vanematel Arduino UNO-del need pesad puuduvad ning prototüüpimise laiendusplaat, mida käesolevas töös kasutatakse, ei võimalda neid pesasid kasutada. Arduino UNO-l on sellele vastavad pesad analoogsisend 4 (SDA) ning analoogsisend 5 (SCL) [49].



**Joonis 13.** I2C tüüpi LCD-tähtekraani ühendamine Arduinoga [50]

I2C tüüpi LCD-tähtekraanile teksti kuvamiseks tuleb lehel [BitBucket](#) alla laadida sellele vastav teek, kuna seda Arduino IDE-s eelnevalt installeeritud pole [51]. Teekide kasutamine teeb programmeerimise lihtsamaks ning programmi lühemaks ja loetavamaks, kuna saab kasutada juba varem valmis kirjutatud koodi. Uute teekite kasutamisest on rohkem kirjutatud punktis 1.5.

Lisaks allalaaditud teegile *LiquidCrystal\_I2C.h* tuleb kasutada ka teeki *Wire.h*, mis on Arduino IDE-s juba olemas. Teekide kasutamiseks tuleb enne meetodeid *setup()* ja *loop()* kirjutada *#include <Wire.h>* ja *#include <LiquidCrystal\_I2C.h>*. LCD-tähtekraani teek võib minna konflikti teise LCD-tähtekraani teekidega, mis on eelnevalt Arduino IDE-s olemas. Konfliktide tekkimisel tuleb kustutada teised teegid ära või paigutada teise kausta. Teegid asuvad Windowsis kaustas *C:/Program Files (x86)/Arduino/libraries*.

Enne *setup()* meetodit kirjeldatakse ära veel LCD-tähtekraani pesad reaga *LiquidCrystal\_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE)* ning määratakse pesa, kust pinget lugema hakatakse (*const int analogInPin = A0;*). Meetodis *setup()* initsialiseeritakse LCD-tähtekraan 16 veeru ja 2 reaga (*lcd.begin(16, 2);*). Meetodis *loop()* seatakse kursor LCD-tähtekraanil kohale (0, 0) ning käivitatakse meetod *readVoltage()*, mis on defineeritud allpool. Meetodi defineerimiseks tuleb määrata tagastustüüp, mis siin on *void*, kuna midagi tagastada ei tule. Meetodi sisu kirjutatakse loogeliste sulgude vahele. Meetodi sees kirjutatakse LCD-tähtekraanile pinge väärtus, mis arvutatakse joonisel 12 oleva valemiga, kus sensori väärtus

leitakse sisseehitatud meetodi *analogRead()* abil. Seejärel oodatakse 2 sekundit, et uuesti väärtust leida ja LCD-tähtekraanile kirjutada (*delay(2000)*) [49].

Eelnevalt kirjeldatud programm on täielikult välja toodud joonisel 14.

```
#include <Wire.h> // Arduino IDE-s olemas
#include <LiquidCrystal_I2C.h> // Tuleb eraldi installida
//https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads

// Ühendab LCD-tähtekraani õigete pesadega ära
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

// Määrab ära, millisest pesast pinge väärtus loetakse
const int analogInPin = A0;

void setup() {
  lcd.begin(16, 2); // Initsialiseerib ekraani 16 veeru ja 2 reaga
}

void loop() {
  lcd.setCursor(0, 0); // Seab kursori kohale (0, 0) - vasakul üleval
  readVoltage(); // Käivitab meetodi readVoltage()
}

void readVoltage() {
  sensorValue = analogRead(analogInPin); // Loeb pesast väärtuse
  // Teeb vajaliku arvutuse ning väljastab selle ekraanile
  lcd.print(sensorValue / 1023.0 * 5.0);
  lcd.print("V"); // Väljastab ekraanile "V"
  delay(2000); // Ootab 2 sekundit
}
```

**Joonis 14.** Pinge väärtuse LCD-tähtekraanile kirjutamise programm

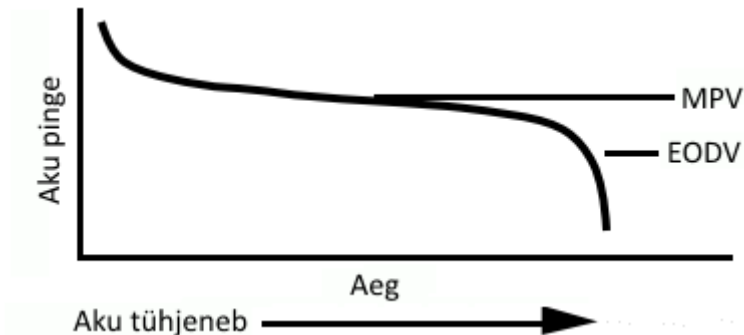
Nüüd on saavutatud hetk, mil teatud aja tagant mõõdetakse aku pinget ning prinditakse LCD-tähtekraanile. Järgmiseks tuleb tuvastada õige hetk, millal aku on piisavalt tühi, et liikuda laadimispunkti ning seal laadima hakata.

## 2.4 Laadima minemise hetke tuvastamine

Tuleb tuvastada hetk, millal aku on piisavalt tühi, et seda peaks laadima, ning samas piisavalt täis, et robot jõuaks liikuda laadimispunkti õigeks ajaks.

Taaslaetavatel akudel langeb pinge eksponentsiaalselt. Nagu jooniselt 15 näha, langeb pinge kiiremini kohe pärast täis laadimist ning seejärel langeb suhteliselt aeglaselt. MPV - keskpunkti pinge (ingl *k mid-point voltage*) on pinge akul, mis on pooleldi tühi. Mõni hetk

pärast seda tuleb järsk pinge langemine, millele järgneb EODV - tühjenemise lõpu pinge (ingl k *end of discharge voltage*). See on hetk, kus aku pole veel päris tühi, kuid paljud seadmed enam ei tööta, et vältida liigset aku tühjenemist [52,53].



**Joonis 15.** Aku pinge vähenemine. MPV - keskpunkti pinge (ingl k *mid-point voltage*). EODV – tühja aku pinge (ingl k *end of discharge voltage*) [52]

Seega tuleks leida aku pinge väärtus, mis on natuke pärast MPV-punkti ning enne EODV-punkti. Väärtus leitakse tehes robotiga katseid nii, et pinge poleks laadima minnes liiga väike, kuna siis ei jõua robot õigeaks ajaks laadimispunkti, kuid samas ka mitte liiga suur, kuna siis liigutakse laadimispunkti liiga vara. Antud töös kasutatud täislaetud aku pinge on 5 volti. Laadimispunkti hakkab robot minema, kui aku pinge on 3.6 volti. Alla 3.5 voldi liigub robot edasi väga aeglaselt. Kuna aku väga kiiresti ei tühjene, on 3.6 volti piisav, et jõuda laadimispunkti enne aku tühjaks saamist.

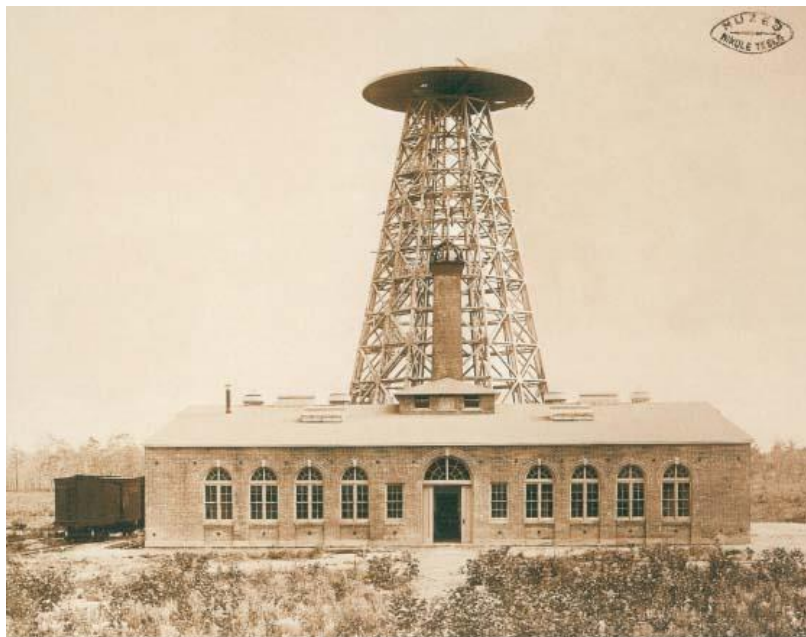
Pärast aku tühjenemise tuvastamist peab robot liikuma laadimispunkti. Järgmises peatükis antakse ülevaade juhtmevabast laadimisest, selle tehnoloogiast ning tutvustatakse Qi juhtmevaba laadimise standardit.

### 3. Juhtmevaba laadimine

Laadimispunkti jõudmisel on robot valmis laadimiseks. Robot hakkab laadima juhtmevabalt. Klassikaline kontaktühendus ei ole väga mugav ning ei võimalda seadmete veekindlust. Antud peatükis antakse ülevaade juhtmevabast laadimisest, selle tehnoloogiatest ning tutvustatakse Qi juhtmevaba laadimise standardit.

#### 3.1 Juhtmevaba laadimise ajalugu

Juhtmevaba energia edastamine sai alguse XX sajandi alguses. 1901. aastal alustas Nikola Tesla Wardenclyffe masti (vt joonis 16) ehitamist. Torni kasutati ringhäälinguks, juhtmevabaks suhtluseks ning juhtmevaba energia edastamiseks. Suurte elektriväljade tõttu polnud voolu edastamine väga efektiivne ning leutis osutus ebaedukaks. Tehnoloogia jäi puutumatuks mitmekümneks aastaks kuni mikrolainete tehnoloogia tekkimiseni [54,55].



**Joonis 16.** Wardenclyffe mast [56]

Pärast Teist maailmasõda hakati jälle uurima juhtmevaba energia edastamist. Tekkisid suure võimsusega elektronlambid, mis suutsid töötada mikrolainete sagedusel ning seega oli võimalik saata energia kiiri (ingl. k *beam of power*) kaugemale [55].

1963. aastal demonstreeris William C. Brown mikrolainete juhtmevaba energia edastamise süsteemi ning aasta hiljem näitas mudellennukit, mis sõitis juhtmevaba energia jõul [55].

1980ndatel aastatel tehti mikrolainete energia edastamisega palju katseid Jaapanis. Hiroshi Matsumoto Kyoto Ülikoolist viis läbi katse, mis näitas mikrolainete energia edastamist läbi ionosfääri [57].

1990ndatel aastatel tekkisid esimesed elektrilised hambaharjad tootja *Braun* poolt, mis kasutasid magnetilist induktsiooni [58].

2008. aastal asutati ühing *Wireless Power Consortium* (WPC), et leida universaalne lahendus mobiiliseadmete juhtmevabale laadimisele. 2010. aastal tutvustati Qi juhtmevaba laadimise standardit, mille kohta saab lähemalt lugeda punktis 4.5.

Üks esimesi seadmeid igapäevaelus, mis kasutab juhtmevaba laadimist, on elektrooniline hambahari, kuid see ei kasutanud Qi juhtmevaba laadimise standardit [59].

Järgmises punktis tutvustatakse lähemalt erinevaid juhtmevaba laadimise tehnoloogiaid.

### **3.2 Juhtmevaba laadimise tööpõhimõte**

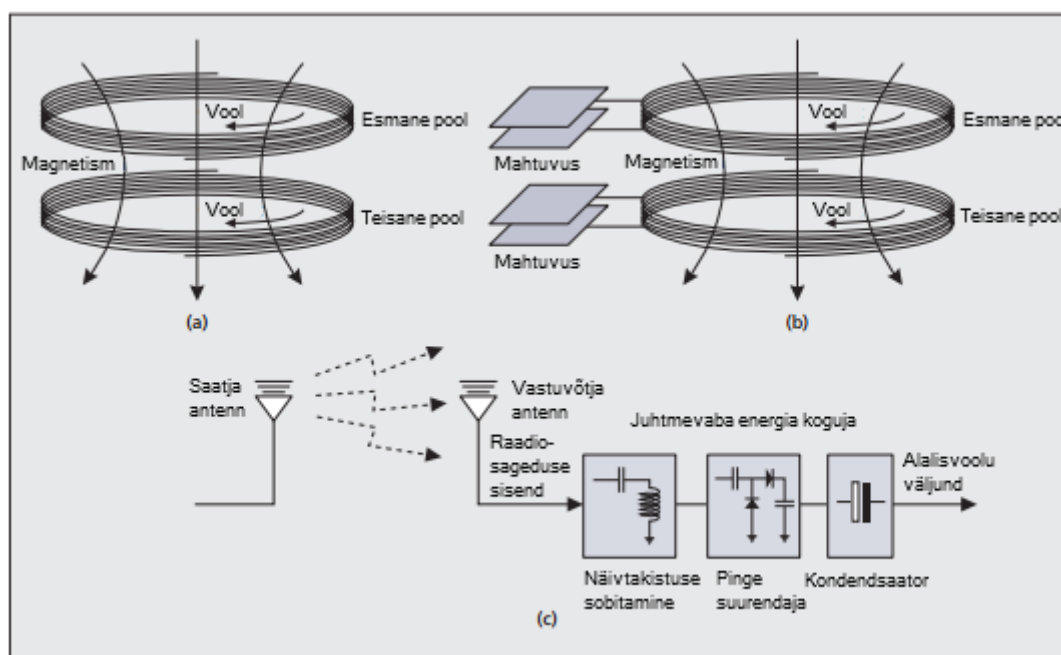
Juhtmevaba laadimise enamkasutatavad tehnoloogiad on: magnetiline induktsioon, magnetiline resonants ning mikrolainete kiirgus. Magnetilise induktsiooni ja -resonantsi puhul mõjub elektromagnetiline väli saatjale lähedal olevatele objektidele; mikrolainete kiirgus mõjub ka kaugel olevatele objektidele (vt tabel 3) [60].



**Tabel 3.** Juhtmevaba laadimise tehnoloogiate võrdlus [54]

Juhtmevaba laadimise tehnoloogia	Eelised	Puudused
Magnetiline induktsioon	Suure võimsusega	Vajab õiget joondumist, saatja ja vastuvõtja peavad olema lähestikku
Magnetiline resonants	Ei nõua joondatust	Saatja ja vastuvõtja peavad olema küllaltki lähestikku
Mikrolainete kiirgus	Efektiivne edastamine kaugel kaugustel	Nõuab vaatesuunalisust (ingl. k <i>line-of-sight</i> )

Jooniselt 17 on näha, et magnetiline induktsioon (vt joonis 17 (a)) ja magnetiline resonant (vt joonis 17 (b)) on tööpõhimõttelt sarnased.



**Joonis 17.** Juhtmevaba laadimise süsteemide mudelid: (a) magnetiline induktsioon; (b) magnetiline resonants; (c) kaugvälja juhtmevaba laadimine [60]

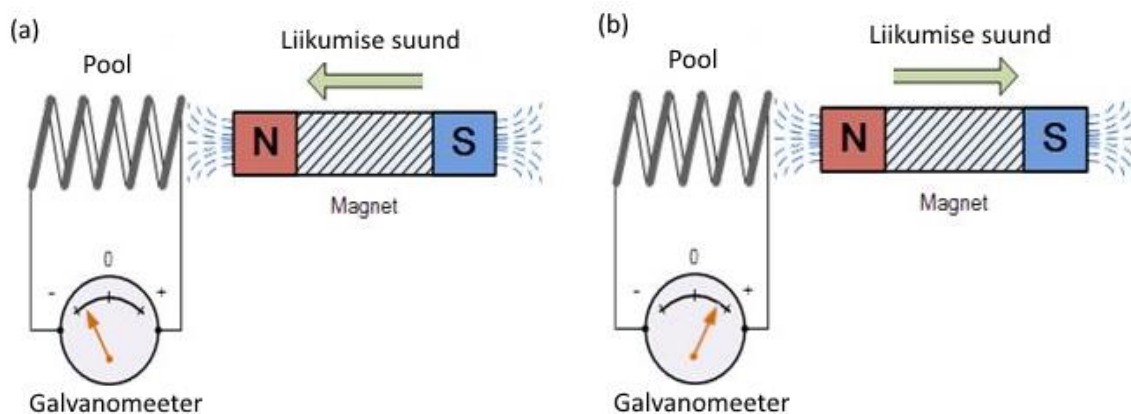
Käesolevas töös keskendutakse juhtmevabale laadimisele, mis põhineb magnetilisele induktsioonile ja -resonantsile. Magnetilist induktsiooni tutvustatakse lähemalt järgmises punktis.

### 3.3 Magnetiline induktsioon

Magnetilise induktsiooni avastamisele andis suure panuse Michael Faraday.

1831. aastal tegi Michael Faraday katseid magnetilise induktsiooniga. Ta tahtis näidata, et magnetismist saab genereerida elektrit. Ta kasutas oma katses püsimagnetit, pooli ja galvanomeetrit. Faraday avastas, et elektromotoorjõud indutseeritakse poolis, kui muutub sellega seotud magnetvoog [61].

Eksperimendis ühendatakse pool galvanomeetriga. Galvanomeetrit kasutatakse elektrijuhis elektrivoolu olemasolu, suuna ja tugevuse määramiseks. Katse alguses on püsimagnet paigal ning galvanomeetri nõel on 0-positsioonil, s.t elektrivoolu pole. Kui magnetit poolile lähemale liigutada, kaldub galvanomeetri nõel ühele poole (vt joonis 18 (a)). Kui magneti liigutamine lõpetada, liigub galvanomeetri nõel tagasi 0-positsiooni. Kui magnet tagasi poolilt eemale liigutada, liigub galvanomeetri nõel teises suunas (vt joonis 18 (b)) [62,63].



**Joonis 18.** Michael Faraday induktsiooni eksperiment magneti, pooli ja galvanomeetriga [62]

Elektrivool võib tekkida ka teistel viisidel: magnetvälja tugevuse muutmisel, magneti poolile lähemale- ja kaugemale liigutades, magneti pooli sisse ja välja liigutades, pooli pööramisel magneti suhtes jne [64].

Magnetilist induktsiooni saab kasutada ka voolu edastamiseks ühest seadmest teise. Saatja seadmes olevas poolis tekitatakse vahelduvvooluga magnetväli. Magnetvälja muutumisel tekib vastuvõtja poolis elektrivool, mida saab kasutada seadme laadimiseks. Voolu edastamiseks peavad saatja pool ja vastuvõtja pool olema üksteisele lähedal.

Magnetilise induktsiooni ja -resonantsi abil voolu edastamine on analoogne. Magnetilist resonantsi tutvustatakse lähemalt järgmises punktis.

### **3.4 Magnetiline resonants**

Magnetilise resonantsi tehnoloogial energia edastamine sarnaneb magnetilise induktsiooni tehnoloogiale.

Süsteem koosneb samuti saatja- ja vastuvõtja poolist. Pooli diameeter ja poolidevaheline kaugus on sarnane. Magnetilise resonantsi tehnoloogial põhinev süsteem töötab vastuvõtja resonantssagedusel. Tavaliselt on magnetilise resonantsi tehnoloogial energia edastamine vähem efektiivne, kui magnetilise induktsiooni tehnoloogial energia edastamine [65,66].

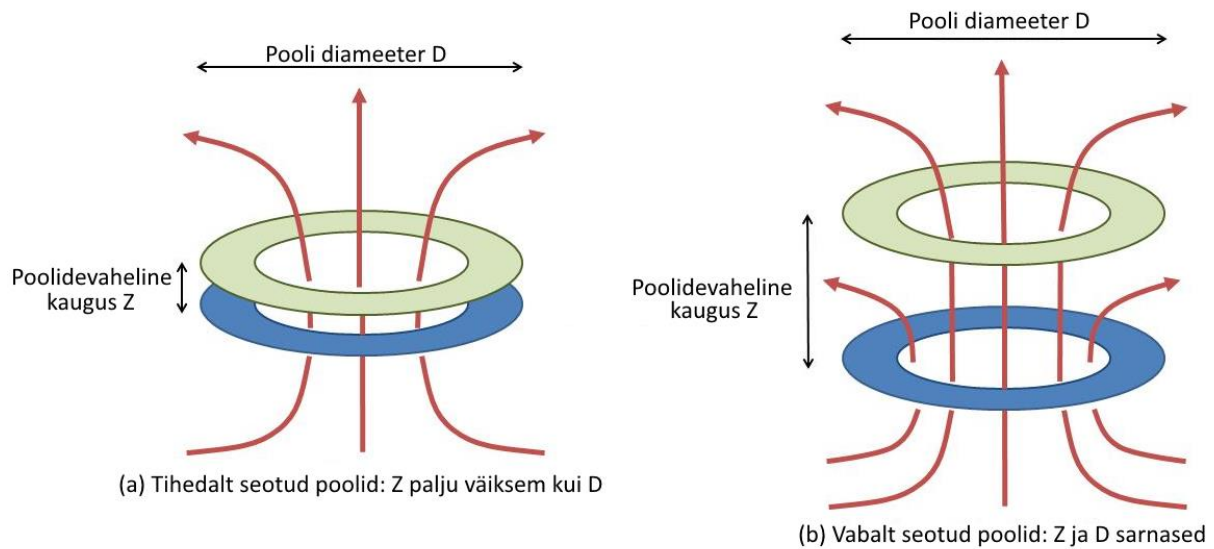
Magnetilise induktsiooni ja -resonantsi põhimõttel töötab ka Qi juhtmevaba laadimise süsteem [67].

### **3.5 Qi juhtmevaba laadimise standard**

Qi on ühingu *Wireless Power Consortium* (WPC) arendatud juhtmevaba laadimise standard [60]. Qi kasutab laadimiseks nii magnetilist induktsiooni kui ka magnetilist resonantsi.

Induktiivse laadimise tehnoloogial liigub laeng läbi saatja ja vastuvõtja pooli, mille vahe on vähem kui 7mm. Üldjuhul peavad olema seadmed joondatud, kuid mõned saatjad kasutavad mitut pooli, mistõttu pole see vajalik [65]. Parim tulemus saavutatakse, kui saatja sagedus on veidi erinev Qi vastuvõtja resonantsisagedusest. Qi saatjad kasutavad tihedalt seotust saatja ja vastuvõtja pooli vahel. See tähendab, et poolid on samade mõõtmetega ning poolide vaheline kaugus on palju väiksem poolide diameetritest (vt joonis 19 (a)) [66].

Saatja ja vastuvõtja pooli kauguse suurenedes väheneb magnetiline sidestus. Madala magnetilise sidestusega süsteemid töötavad vastuvõtja resonantsisagedusel [66]. Juhtmevaba laadimise resonantsirežiimis ei pea seadmed olema üksteisele niivõrd lähedal kui induktiivrežiimis. Saatja saab tuvastada ning laadida Qi vastuvõtjaid kuni 45mm kauguselt. Seetõttu saab saatja peita ka erinevate esemete sisse või kinnitada vastuvõtja sügavamale seadme sisse [65]. Resonantsirežiimis on pooli diameeter ja poolide vaheline kaugus sarnased (vt joonis 19 (b)) [66].



**Joonis 19.** a) Tihedalt seotud poolid; b) Vabalt seotud poolid [66]

Qi juhtmevabad laadijad saavad opereerida nii induktiivrežiimis kui ka resonantsirežiimis. Režiim valitakse saatja ja vastuvõtja poolide vahelisele kaugusele vastavalt [66].

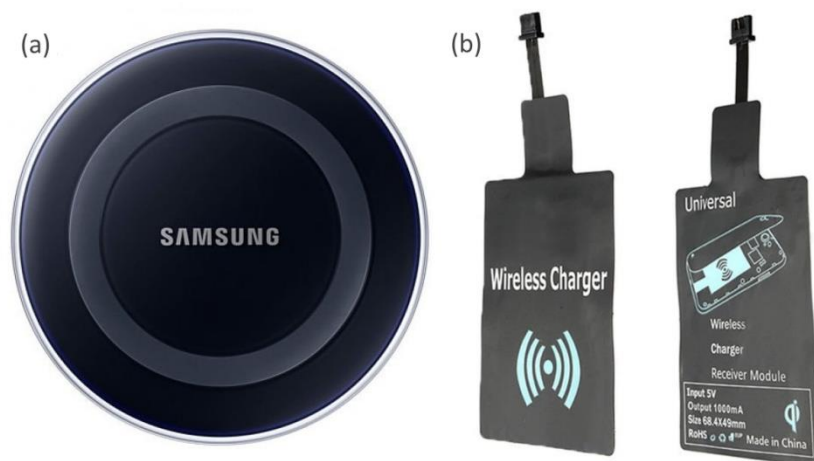
Antud töös kasutatakse samuti Qi standardile vastavat juhtmevaba laadimise vastuvõtjat ning saatjat.

### 3.6 Aku laadimine juhtmevabalt

Laadimiseks kasutatakse käesolevas töös juhtmevaba laadijat ja vastuvõtjat. Mõlemad vastavad Qi standardile.

Saatja on Samsungi toodetud ning sellega saab laadida kõiki Qi standardile vastavaid seadmeid (vt joonis 20 (a)). Laadija tuleb ühendada USB-juhtme abil seinalaadijaga või arvutiga. Laadijal on sisse ehitatud LED-tuled, mis on laadimise ajal sinist värvi ning kui seade on täis laetud või seade pole laadijaga ühenduses, on LED-tuled rohelist värvi [68].

Laadija vastuvõtja sobib kõigile seadmetele, millel on Micro-USB pesa (vt joonis 20 (b)). Kuna töös kasutatakse akupanka, mida laetakse Micro-USB-juhtmega, sobib see vastuvõtja akupanga laadimiseks.



**Joonis 20.** Juhtmevaba laadija (a) ja juhtmevaba laadija vastuvõtja (b) [68,69]

Laadimiseks tuleb ühendada juhtmevaba laadija seinalaadijaga ning akupank ühendada vastuvõtjaga. Kui vastuvõtja panna laadija vastu, hakkab akupank laadima. Laadimise ajal on laadijas olevad LED-tuled sinised ning akupangas olevad LED-tuled punased (vt joonis 21).



**Joonis 21.** Laadijas olevad LED-tuled on sinised ning akupangas olevad LED-tuled on punased

Käesolevas peatükis anti ülevaade juhtmevabast laadimisest ning tutvustati erinevaid juhtmevaba laadimise tehnoloogiaid. Tutvustati Qi standardit ning antud töös laadimiseks kasutatud seadmeid. Järgmises peatükis kirjeldatakse, kuidas jõuab robot laadimispunkti, kus hakkab juhtmevabalt laadima.

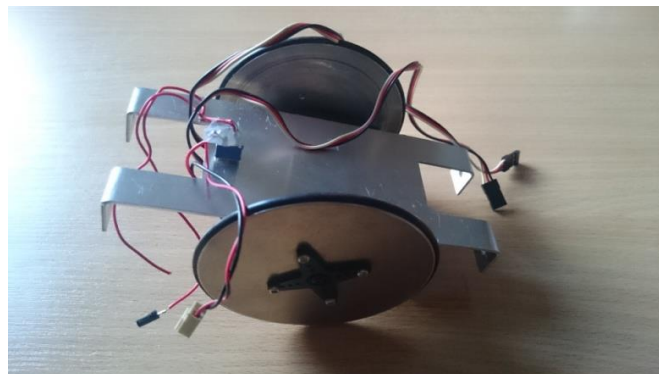
## 4. Laadimispunkti liikumine

Arduino arendusplaat saab nüüd aru, et aku on tühjenemas ning tuleb liikuda laadimispunkti. Selleks tuleb Arduino ühendada ratastega, mida liigutavad servomootorid ning seejärel infrapuna sensorite abil tuvastada laadimispunkti asukoht, et laadima minna.

### 4.1 Arduino ühendamine mootoritega

Selleks, et Arduino laadimispunkti jõuaks, on vaja Arduinole külge ühendada servomootorid.

Käesolevas töös kasutatakse juba valmis tehtud kahe rattaga alust, millele saab peale paigutada Arduino arendusplaadi (vt joonis 22).



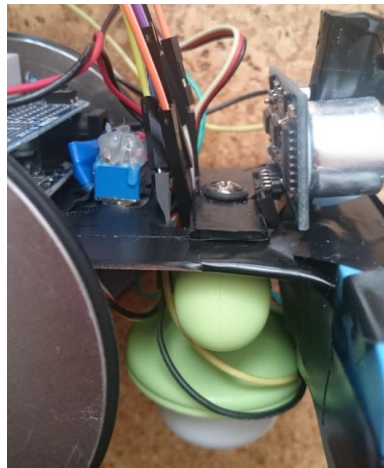
**Joonis 22.** Kahe rattaga alus robotile

Rattaid juhivad kaks servomootorit, mille saab Arduinoga ühendada. Kuna kahe rattaga ei ole robot tasakaalus, on lisatud ette veel kolmas ratas. Kuna see ei ole mootoriga ühendatud, on oluline, et see liiguks kergesti igas suunas. Seetõttu on ratta alumises osas kuul. Ratas selle töö jaoks on ostetud poest Tiger. Et ratas paremini fikseeritud oleks, on ratta plastmassist osal (vt joonisel 23 roheline osa) ümarad kohad lamedamaks lihvitud.



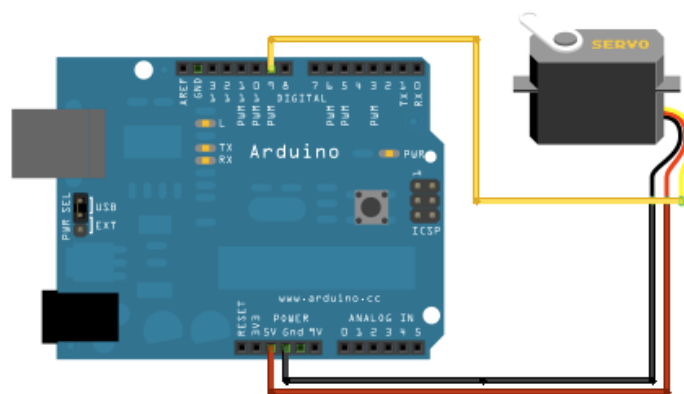
**Joonis 23.** Lihvitud plastmassist osaga ratas

Ratta roboti aluse külge kinnitamiseks on kasutatud isoleerteibiga kaetud alumiiniumist plaati. Plaadi ja ratta plastmassist osa sisse on tehtud augud. Ratas on roboti alusega ühendatud nii, et alus on alumiiniumplaadi ja ratta vahel ning plaat on ühendatud ratta plastmassist osaga kruvi abil (vt joonis 24)



**Joonis 24.** Ratta ühendamine robotiga

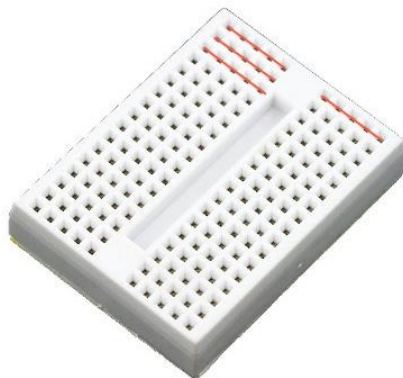
Servomootori ühendamiseks Arduinoga on vaja kolme juhet: VCC (toide), GND (maandus) ja signaal. VCC juhe on punane, GND must ning signaal kollane (vt joonis 25) [70].



**Joonis 25.** Arduino ühendamine servomootoriga [70]

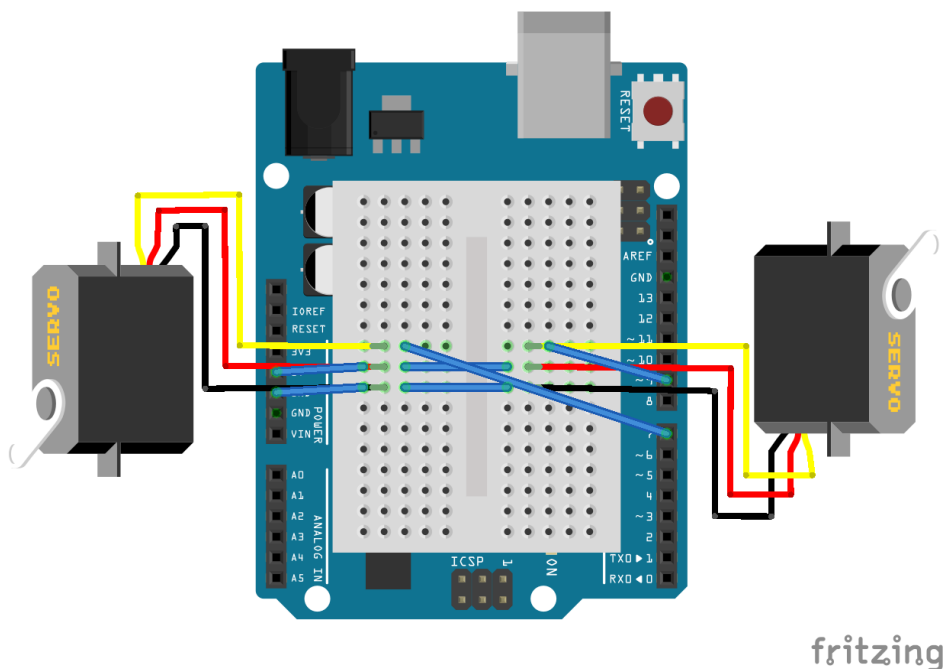
Ühendada on vaja kaks servomootorit. Arduino arendusplaadil pole selleks piisavalt pesasid. Puudu on üks VCC (joonisel 25 on selle pesa tähis 5V). Signaali edasi kandmiseks kasutame prototüüpimise laiendusplaati, millega saab Arduino peale lisada pisikese maketeerimislauda.

Maketeerimislaual on rajad ühenduses maketeerimislaua keskel oleva osa suhtes risti. Signaali katkestab maketeerimislaua keskel olev osa (vt joonis 26).



**Joonis 26.** Signaali kandumine maketeerimislaual (punased jooned joonisel) [71]

Kuna maketeerimislaua keskel olev osa katkestab signaali, on joonisel 27 toodud VCC (punane) ja GND (must) signaal maketeerimislaua vasakult poolelt üle paremale poolele. Vasakpoolne servomootor on ühendatud pesaga 7 ning parempoolne pesaga 9. Neid kasutame hiljem programmi kirjutamises servomootorite ühendamiseks, et saaks neid vajadusel liigutada.



**Joonis 27.** Kahe servomootori ühendamine Arduino maketeerimislaua ja arendusplaadiga



Servomootorite liigutamiseks tuleb kirjutada programm, mis tuvastab servomootorid ning neid vajadusel liigutab.

## 4.2 Mootorite liigutamine

Servomootorite kasutamiseks tuleb kasutada välist teeki, mis on Arduino IDE-s juba olemas. Selleks tuleb valida ülemisest menüüst *Visand*, seejärel *Include Library* ning sealt valida *Servo*. Nüüd peaks programmi tekkima rida `#include <Servo.h>`. Enne `setup()` funktsiooni kirjutamist defineerime kaks *Servo* objekti, `servoLeft` ja `servoRight`. Selleks tuleb kirjutada lihtsalt read `Servo servoLeft` ning `Servo servoRight`. Nüüd hakkame `setup()` funktsiooni kirjutama.

Selleks, et programm teaks, milliste füüsilise pesadega servomootorid ühendatud on, tuleb kirjutada read `servoLeft.attach(7)` ning `servoRight.attach(9)`. Nüüd on servomootorid ühendatud ning neid saab hakata liigutama.

Antud töös kasutatakse 360-kraadi servomootoreid. Nende liigutamiseks tuleb *servo* objekti kirjutada arv vahemikus 0-180. Arv 0 tähendab, et mootor liigub ühes suunas täiskiirusel, arvuga 180 liigub teises suunas täiskiirusel ning 90 tähendab, et mootor jääb seisma. Arvu 90 asemel võib olla sellest veidi erinev arv ning seda tuleb lihtsalt proovida. Antud töös kasutatud servomootoritel on selleks 93 [72].

Servomootorite katsetamiseks tuleb teha programm, mis paneb roboti ruudus sõitma (vt joonis 28). Selleks tuleb sõita natuke aega edasi, siis pöörata ning korrata tegevust. Edasi sõitmiseks, pööramiseks ja roboti peatamiseks tuleb teha eraldi meetodid, mida hakata meetodis `loop()` välja kutsuma. Edasi sõitmiseks tuleb parem servomootor panna asendisse 180 ning teine asendisse 7, et mõlemad mootorid liiguksid samal kiirusel. Ka neid arve tuleb katsetada, et robot liiguks otse. Paremale pööramiseks tuleb mõlemad servomootorid panna asendisse 180, kuna servomootorid on ühendatud Arduinoga nii, et üks on suunaga roboti ette ja teine taha. Roboti peatamiseks tuleb mõlemad servomootorid panna asendisse 93.

Asendi määramiseks servomootoril tuleb kirjutada vasaku servomootori puhul rida `servoLeft.write(x)`, kus `x` on asendi väärtus, ning analoogne parema servomootori jaoks. Pärast asendite määramist tuleb hetkeks programm peatada reaga `delay(x)`, kus `x` on peatamise aeg millisekundites. Programmi peatamine on vajalik selleks, et programm ei

läheks edasi uue käsu täitmisega enne, kui eelmine on täidetud. Näiteks paremale pöörates tuleb hoida servomootorites asendeid seni, kuni on keeratud 90 kraadi. Selleks kuluva aja saab teada katsetades. Antud töös kasutatud servomootorite puhul on selleks 400 millisekundit.

Edasi sõitmise meetod (*forward(int n)*) ja paremale sõitmise meetod (*turnRight(int n)*) võtavad argumendiks täisarvu (*int n*), mitu millisekundit tuleb oodata enne järgmise käsu täitmist. Meetodi välja kutsumisel antakse meetodile kaasa argument kujul *forward(2000)*. Meetodi sees asendatakse *delay(n)* sees olev *n* argumendi väärtusega (2000). Meetodite loomisest on rohkem räägitud peatükis 2.3.

Pärast kolme meetodi loomist saab need meetodis *loop()* välja kutsuda. Ruudus hakkab robot sõitma nii, et meetodid *forward(int n)*, mis paneb roboti edasi liikuma, ja meetod *turnRight(int n)* käivitatakse 4 korda järjest *for*-tsükli abil. Seejärel peatatakse robot meetodi *stopRobot()* abil programmi töö kaheks sekundiks ning siis käivitatakse meetod *loop()* uuesti.

```

#include <Servo.h> // Arduino IDE-s olemas

Servo servoLeft;    // Loob servo objekti
Servo servoRight;   // Loob servo objekti

void setup() {
    servoLeft.attach(7); // Ühendab servo objekti pesaga 7
    servoRight.attach(9); // Ühendab servo objekti pesaga 9
}

void loop() {
    // Tsükkel, mida läbitakse 4 korda
    for (int i = 0; i < 4; i++) {
        forward(2000); // Käivitab forward() meetodi
        turnRight(400); // Käivitab turnRight() meetodi
    }
    stopRobot(); // Käivitab stopRobot() meetodi
    delay(2000); // Peatab programmi töö kaheks sekundiks
}
// Robot liigub edasi
void forward(int n) {
    servoLeft.write(180);
    servoRight.write(7);
    delay(n);
}
// Robot pöörab paremale
void turnRight(int n) {
    servoLeft.write(180);
    servoRight.write(180);
    delay(n);
}
// Robot peatub
void stopRobot() {
    servoLeft.write(93);
    servoRight.write(93);
}

```

**Joonis 28.** Roboti ruudus sõitmise programm

Robot suudab nüüd liikuda. Järgmiseks peab robot leidma laadimispunkti asukoha. Selleks kasutatakse infrapuna sensoreid, mida tutvustatakse järgmises punktis.

### 4.3 Laadimispunkti asukoha tuvastamine infrapuna sensorite abil

Infrapunakiirgust kasutatakse igapäevaelus kaugjuhtimispultides, näiteks televiisori pultides. Vastuvõtja on televiisoris ning saatja on puldis. Saatja saadab televiisoris olevale vastuvõtjale signaali ning televiisor reageerib vastavalt sellele.

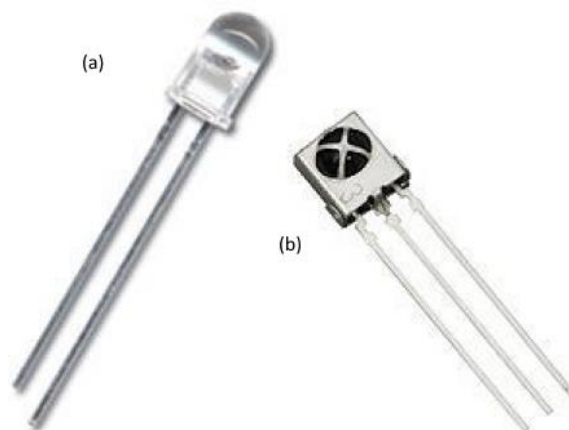
Antud töös kasutatakse infrapuna sensoreid kahel Arduino arendusplaadil. Üks Arduino on roboti peal ning teine laadimispunktis. Laadimispunktis oleva Arduino peal on infrapuna saatja ning robotil oleva Arduino peal on infrapuna vastuvõtja.

### 4.3.1 Infrapunakiirgus

Infrapunavalgus ehk infrapunakiirgus on valgus, mis pole inimese silmale nähtav. Infrapunakiirgust leidub ka looduses. Kõik, mis kiirgab soojust, kiirgab ka infrapunakiirgust. Selleks, et looduslik infrapunakiirgus ei segaks infrapunakiirgust kasutavaid seadmeid, on saatja signaal moduleeritud teatud kandevasagedusele. Tavaliselt kasutatakse sagedust 38kHz. See tähendab, et signaal lülitatakse sisse ja välja 38 000 korda sekundis [73].

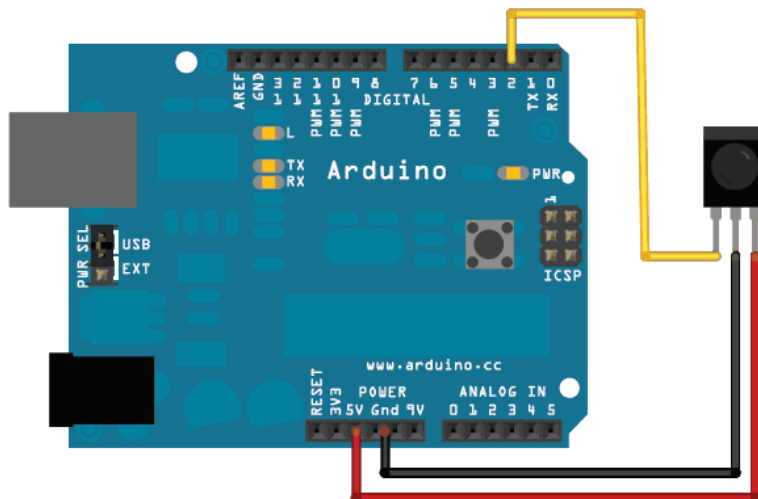
### 4.3.2 Infrapuna sensorite ühendamine Arduinoga

Laadimispunktis oleva Arduino peal on infrapuna saatja (vt joonis 29 (a)) ning robotil oleva Arduino peal on infrapuna vastuvõtja (vt joonis 29 (b)).



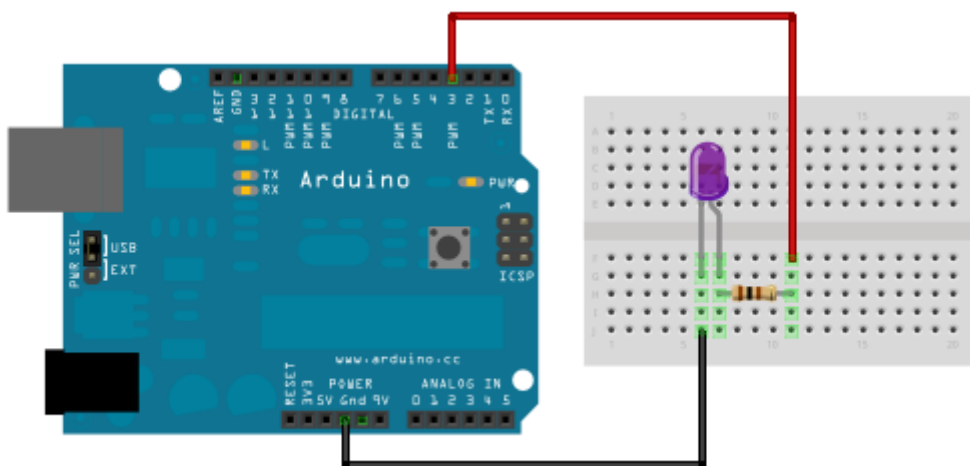
**Joonis 29.** Arduino infrapuna saatja (a) ja -vastuvõtja (b) [74]

Infrapuna vastuvõtja ühendamiseks tuleb ühendada kolm juhet: signaal (kollane), baas (must) ja toide (punane) (vt joonis 30). Juhtmete järjekord sõltub sensorist [75].



**Joonis 30.** Infrapuna vastuvõtja ühendamine Arduinoga [76]

Infrapuna saatja ühendamiseks tuleb ühendada kaks juhet: baas (must) ja toide (punane) (vt joonis 31). Toitega tuleb ühendada sensori pikem jalg. Toite ja sensori vahele tuleb ühendada ka takisti, et piirata valgusdiodi läbivat voolu. Antud töös kasutatakse takistit, mille takistus on  $330\Omega$  [77].



**Joonis 31.** Arduino infrapuna saatja ühendamine [78]

Robotil olevale Arduinole ja laadimispunkti olevale Arduinole tuleb kirjutada programm, et üks andur saadaks teatud aja tagant signaali ning teine andur oskaks seda vastu võtta. Sellega tutvutakse järgmises punktis.

### 4.3.3 Infrapuna sensorite abil laadimispunkti leidmine

Laadimispunkti leidmiseks tuleb kirjutada mõlemale infrapuna sensorile programm. Üks sensor hakkab teatud aja tagant signaali saatma ning teine sensor võtab selle vastu.

Mõlema programmi kirjutamiseks tuleb kasutada teeki, mida Arduino IDE-s olemas ei ole, et infrapuna sensoreid kasutada. Selleks tuleb lehelt [GitHub](#) alla laadida uus teek *Arduino-IRremote* [79]. Uue teegi kasutamiseks tuleb eelnevalt kustutada või tõsta teise kausta Arduino IDE-s olev *RobotIRRemote* teek, kuna vastaval juhul tekivad teekide vahel konfliktid. *RobotIRRemote* teek asub *Windowsis* kaustas *C:/Program Files (x86)/Arduino/libraries*. Uue teegi installimisega tulevad kaasa ka näiteprogrammid, mis on kättesaadavad Arduino IDE-s valides ülemisest menüüst *File*, seejärel *Examples* ning tekkinud menüüst *IRremote*. Uute teekide kasutamisest on rohkem kirjutatud punktis 1.5.

Laadimispunktis oleva Arduino programmeerimiseks kasutatakse näidisprogrammi *IRsendDemo* (vt joonis 32). Näidisprogramm on võetud aluseks ning selle põhjal on tehtud käesolevaks tööks vajalik programm. Meetodist *loop()* on read *irsend.sendSony(0xa90, 12)* ja *delay(40)* tõstetud tsüklist välja ning programmi peatamine on tõstetud 100 millisekundini (*delay(100)*). Nüüd saadab infrapuna saatja iga 100 millisekundi tagant signaali väärtusega a90 (mille väärtus kuueteistkümnendsüsteemis on 0xa90).

```
/*
 * IRremote: IRsendDemo - demonstrates sending IR codes with IRsend
 * An IR LED must be connected to Arduino PWM pin 3.
 * Version 0.1 July, 2009
 * Copyright 2009 Ken Shirriff
 * http://arcfn.com
 */

#include <IRremote.h> // Tuleb eraldi installida

IRsend irsend; // Luuakse IRsend objekt

void setup() {
}

void loop() {
  irsend.sendSony(0xa90, 12); // Saadab signaali, mille väärtus on a90
  delay(100); // Peatab programmi 100 millisekundiks
}
```

**Joonis 32.** Infrapuna saatja programm

Infrapuna vastuvõtja programmi kirjutamiseks võetakse aluseks näiteprogramm *IRrecvDemo* (vt joonis 34). Tekitatakse *irrecv* objekt, mis on võimeline võtma vastu infrapuna signaali. Kui signaal leitakse, tegutsetakse edasi vastavalt signaali väärtusele. Selleks kasutatakse tingimuslauset *switch*, mis saab argumendiks vastuvõetud signaali väärtuse (*results.value*). Tingimuslause *switch* sees on mitu *case*-lauset, mis määravad ära, mida tuleb teha teatud väärtuse korral. Näiteks *case 0xFD8877* blokki liigutakse siis, kui saadakse väärtus 0xFD887. Joonisel 28 olev programm on kirjutatud nii, et puldil (vt joonis 33) olevate noolenuppude vajutamisel liigub robot vastavasse suunda. Selleks kutsutakse iga noolenupu vajutuse korral välja teatud meetod, mis on kirjeldatud pärast meetodit *loop()* (vt joonis 34).



### Joonis 33. Arduino pult [80]

Kui signaali pole kirjeldatud *case*-lausega ning robot on arvutiga ühendatud, väljastatakse signaali väärtus jadapordi monitorile (*default*-lause). Pärast signaali otsimist oodatakse enne uue signaali otsimist 100 millisekundit (*delay(100)*).

```
/*
 * IRremote: IRrecvDemo - demonstrates receiving IR codes with IRrecv
 * An IR detector/demodulator must be connected to the input RECV_PIN.
 * Version 0.1 July, 2009
 * Copyright 2009 Ken Shirriff
 * http://arcfn.com
 */

#include <IRremote.h>
#include <Servo.h> // Arduino IDE-s olemas

Servo servoLeft;    // Loob servo objekti
Servo servoRight;   // Loob servo objekti

int RECV_PIN = 11; // Määrab ära sensori pesa
IRrecv irrecv(RECV_PIN); // Teeb irrecv objekti
```

```

decode_results results; // Teeb results objekti

void setup() {
  servoLeft.attach(7); // Ühendab servo objekti pesaga 7
  servoRight.attach(9); // Ühendab servo objekti pesaga 9
  Serial.begin(9600); // Alustab jadapordi monitori
  irrecv.enableIRIn(); // Alustab infrapuna vastuvõtja
}

void loop() {
  if (irrecv.decode(&results)) { // Kontrollib, kas saab signaali
    // Valib tegevuse vastavalt results.value väärtusele
    switch(results.value) {
      case 0xFD8877: // Kui väärtus on 0xFD8877
        forward(500); // Robot läheb edasi
        stopRobot(); // Robot peatatakse
        break; // Lõpetatakse case plokk
      case 0xFD6897:
        turnRight(400); // Robot pöörab paremale 90 kraadi
        stopRobot();
        break;
      case 0xFD28D7:
        turnLeft(400); // Robot pöörab vasakule 90 kraadi
        stopRobot();
        break;
      case 0xFD9867:
        reverse(500); // Robot läheb tagasi
        stopRobot();
        break;
      default: // Kui tuleb teine signaal
        // Väärtus väljastatakse jadapordi monitorile
        Serial.println(results.value, HEX);
    }
    irrecv.resume(); // Võtab vastu järgmise väärtuse
  }
  delay(100); // Peatab programmi töö 100 millisekundiks
}

// Võtab argumendiks täisarvu ning
// sõidab nii kaua edasi (millisekundites)
void forward(int n) {
  servoLeft.write(180);
  servoRight.write(7);
  delay(n);
}

// Võtab argumendiks täisarvu ning
// sõidab nii kaua tagasi (millisekundites)
void reverse(int n) {
  servoLeft.write(7);
  servoRight.write(180);
}

```



```

    delay(n);
}

// Võtab argumendiks täisarvu ning
// pöörab nii kaua paremale (millisekundites)
void turnRight(int n) {
    servoLeft.write(180);
    servoRight.write(180);
    delay(n);
}

// Võtab argumendiks täisarvu ning
// pöörab nii kaua vasakule (millisekundites)
void turnLeft(int n) {
    servoLeft.write(0);
    servoRight.write(0);
    delay(n);
}

// Peatab roboti
void stopRobot() {
    servoLeft.write(93);
    servoRight.write(93);
}

```

### Joonis 34. Infrapuna vastuvõtja programm

Praegu on programm tehtud nii, et robot reageerib ainult puldil olevate noolenuppude signaalile ja siis liigub neile vastavalt. Kui robot hakkab laadimispunkti asukohta tuvastama, on *switch*-tingimuslause sisu teine. Algoritmi kirjeldatakse rohkem punktis 4.5.

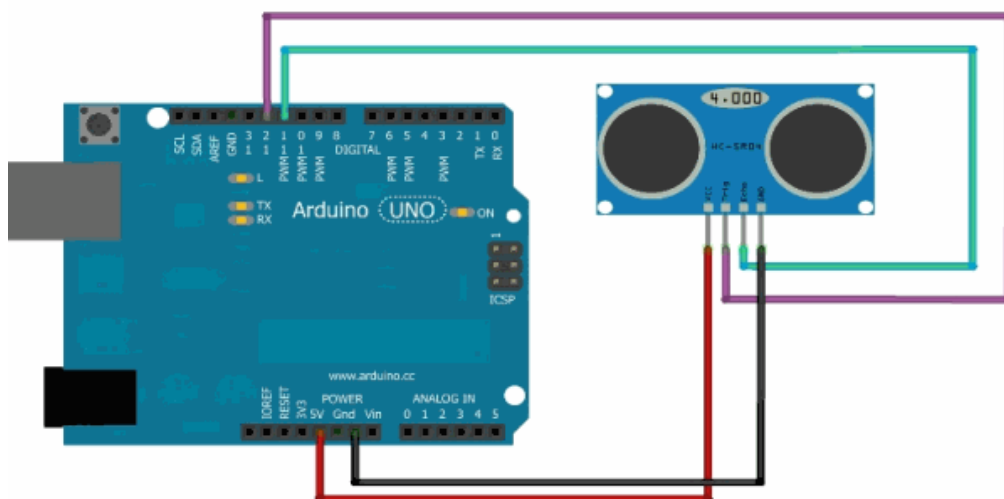
Selleks, et robot liikudes seintele vastu ei sõidaks, on vaja ühendada robotile ultraheli sensor. Sensori ühendamise ja kasutamisega tutvutakse järgmises punktis.

## 4.4 Ultraheli sensor

Ultraheli sensor mõõdab anduri ees oleva vahemaa takistuseni. Liiga väikse vahemaa korral robot enam edasi ei liigu, et vältida kokkupõrget takistusega.

Ultraheli sensor koosneb saatjast ja vastuvõtjast. Saatja saadab heliimpulsi ja vastuvõtja võtab selle vastu. Aeg, mis kulub saatmisest vastuvõtmiseni teisendatakse sentimeetriteks [81].

Ultraheli sensori kasutamiseks tuleb Arduinoga ühendada 4 juhet: toide (punane), baas (must) ja kaks signaali (Trig (lilla) ja Echo (roheline)) (vt joonis 35).



**Joonis 35.** Ultraheli sensori ühendamine Arduinoga [82]

Ultraheli sensori kasutamiseks tuleb kirjutada programm (vt joonis 36). Programmis tuleb esmalt enne *setup()* meetodit määrata ära õiged *Echo* ja *Trig* pesad ning *EchoTime* pikkus, milleks on 0. *EchoPin* väärtus on 12 ning *TrigPin* väärtus 13. Meetodis *setup()* initsialiseeritakse *EchoPin* ja *TrigPin* ning alustatakse jadapordi monitor, et saaks sinna väärtusi väljastada. Meetodis *loop()* saadab *TrigPin* signaali (*digitalWrite(TrigPin, HIGH)*), ootab 10 mikrosekundit ning siis lõpetab signaali saatmise (*digitalWrite(TrigPin, LOW)*). Seejärel mõõdetakse meetodi *pulse()* abil aeg, mis kulub, et *EchoPin* signaali vastu võtaks (*EchoTime = pulseIn(EchoPin, HIGH)*). Seejärel arvutatakse kulunud aeg ümber sentimeetriteks ( $(EchoTime/2)/29$ ) ning väljastatakse väärtus jadapordi monitorile.

```

/*
 * M. Metshein
 * http://metshein.com/index.php/arvuti/arduino/1017-27-arduino-projekt-22-
 * ultraheli-ja-infrapunaandur
 */

// Määrab õigete pesad
int EchoPin = 12;
int TrigPin = 13;
long EchoTime = 0;

void setup() {
  Serial.begin(9600); // Alustab jadapordi monitori
  // Initsialiseerib Echo ja Trig pesad
  pinMode(EchoPin, INPUT);
  pinMode(TrigPin, OUTPUT);
}

void loop() {
  digitalWrite(TrigPin, HIGH); // Trig saadab signaali
  delayMicroseconds(10);
  digitalWrite(TrigPin, LOW); // Lõpetab signaali saatmise

  EchoTime = pulseIn(EchoPin, HIGH); // Echo võtab signaali vastu
  // Kulunud aeg teisendatakse sentimeetriteks
  int dist = (EchoTime/2)/29;
  Serial.print(dist, DEC); // Väljastab jadapordi monitorile
  Serial.println("cm");
  // Ootab 1 sekundi
  delay(1000);
}

```

**Joonis 36.** Ultraheli sensori kasutamise programm [81]

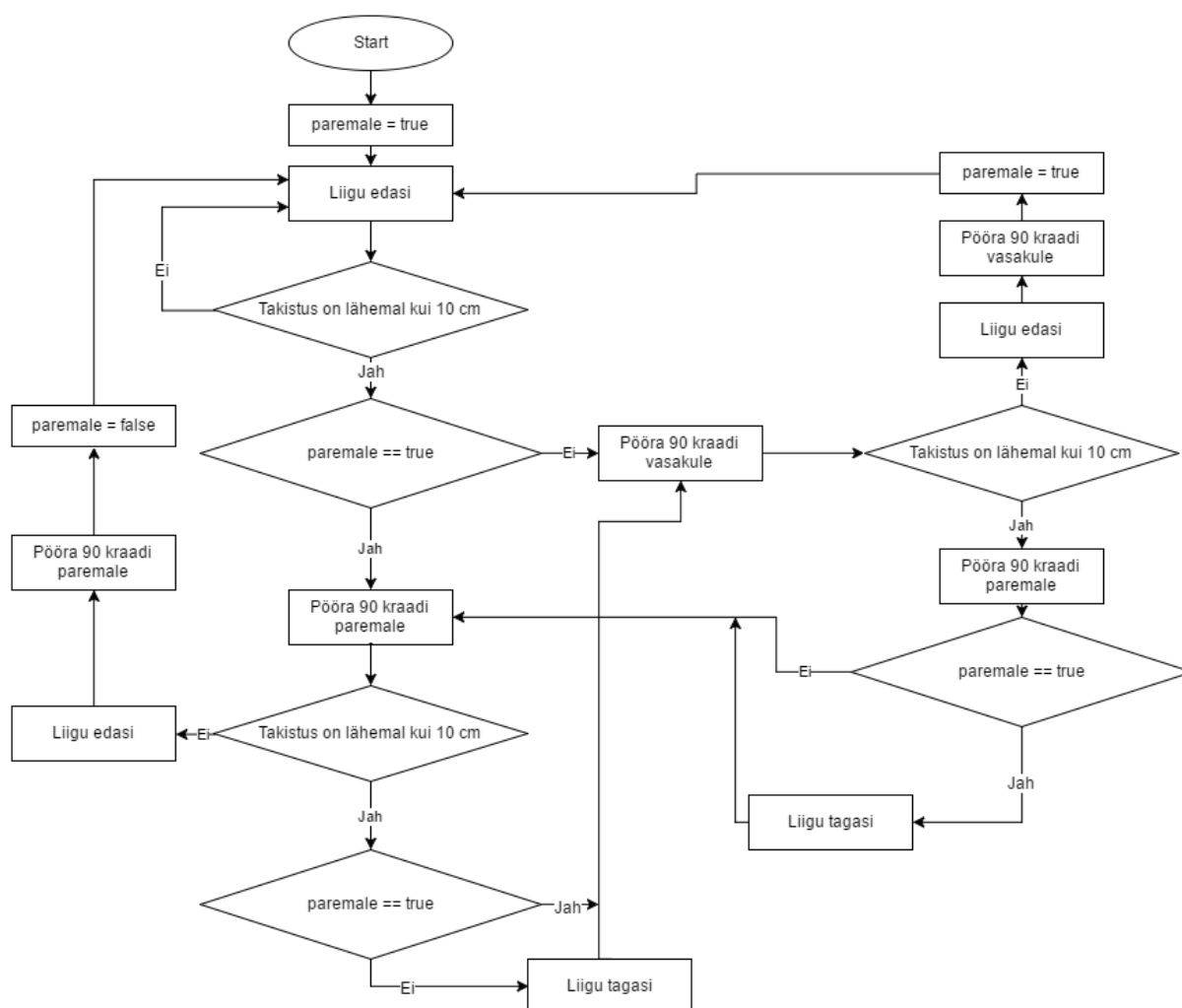
Lõplikus programmis kasutatakse mõõdetud vahemaad eesolevast takistusest, et mitte vastu seina või laadimispunkti sõita. Seda saab kontrollida *if*-tingimuslausega. Vahel annab ultraheli sensor tulemuseks 0, kuigi takistus on tegelikult kaugemal. Programmi kirjutades tuleb sellega arvestada kirjutades *if*-tingimuslause, et kui tulemus on 0, tuleb seda eirata.

Lõplik laadimispunkti liikumise programm pannakse kokku erinevatest koodiplokkidest ning tehakse ka mõningad muudatused. Lõplik programm on välja toodud lisas 1. Laadimispunkti liikumise algoritm on põhjalikumalt kirjeldatud järgmises punktis.

## 4.5 Laadimispunkti liikumise algoritm

Robot liigub toas, kus pole põrandal alla 5 sentimeetrilaiuseid objekte, näiteks juhtmed ja tooliljalad. Alguses liigub robot toas vabalt ringi ning väldib seinte vastu sõitmist. Kui aku on nii tühi, et tuleb laadima minna, hakkab robot laadimispunkti otsima.

Vabas olekus liigub robot toas edasi-tagasi. Algoritm on kirjeldatud joonisel 37 oleva vooskeemiga. Üks kord pöörab paremale ning järgmine kord vasakule. Pööre tehakse siis, kui edasi sõites tuleb takistus ette. Kui pööret tehes tuleb sein ette, pööratakse teisele poole. Võib juhtuda, et mõlemale poole pöörates on sein liiga lähedal. Vältimaks olukorda, kus robot jääbki pöörama, kontrollitakse, kummale poole pidi robot alguses pöörama. Kui mõlemal pool on takistus liiga lähedal, liigub robot natuke tagasi ning proovib uuesti pöörata.



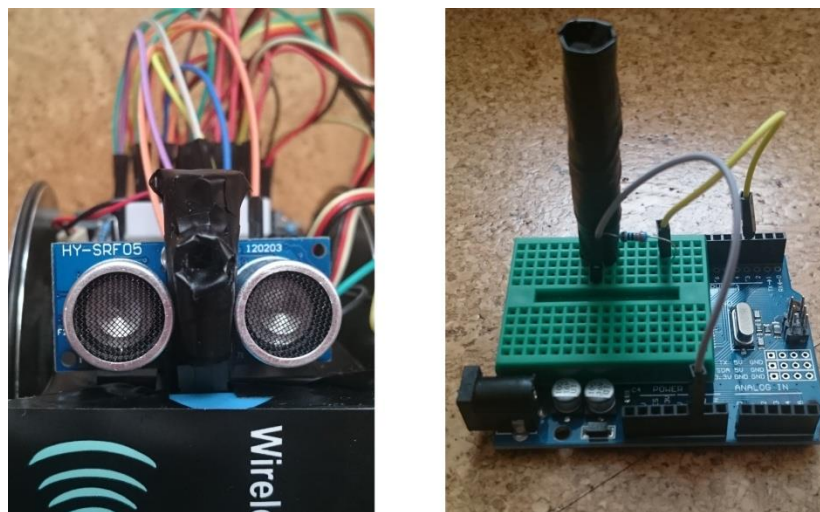
Joonis 37. Roboti vabalt liikumise algoritm

Joonisel 37 olev skeem algab joonise üleval olevast ovaalist "Start". Edasi liigub programm noolte suunas. Ristkülikutes on tegevused ning rombides tingimuslause, kust edasi liigutakse vastavalt sellele, kas tingimuslause oli tõene või väär.

Pärast igat edasiliikumist kontrollitakse aku pinget. Kui aku pinge on nii väike, et tuleb minna laadima, lõpetab robot vabalt liikumise ning hakkab liikuma laadimispunkti poole.

Laadimispunkti liikumise algoritm on sarnane vabalt liikumise algoritmile. Erinevus seisneb selles, et pärast edasiliikumist teeb robot koha peal täisringi, otsides samal ajal laadimispunktist tulevat signaali. Kui sein tuleb ette, pöörab robot samamoodi, nagu vabalt liikudes. Kuna infrapuna vastuvõtja on roboti esiosas, teab robot, et signaali leides tuleb liikuda edasi. Kui signaal on leitud ning takistus on vähem kui 20 sentimeetri kaugusel robotist, liigub robot otse, kuni jõuab laadimispunkti.

Infrapuna saatja ning infrapuna vastuvõtja on ümbritsetud toruga, et signaal leviks teatud suunas ning signaali saaks tuvastada ainult ühest suunast (vt joonis 38). Vastasel korral saab robot signaali valel ajal ning liigub kohta, kus laadimispunkti tegelikult ei ole.



**Joonis 38.** Infrapuna vastuvõtja (vasakul) ja saatja (paremal) ümber olevad torud

Robot jõuab laadimispunkti, kuid alati ei hakka laadima. Laadimiseks peab olema robotil olev vastuvõtja joondatud täpselt juhtmevaba laadijaga kohakuti, kuid vahel ei jõua robot nii täpselt vastu laadijat. Praegu ongi juhtmevaba laadimise suureks probleemiks see, ei on vaja seade laadijaga täpselt joondada. Järgmises peatükis antakse ülevaade käesoleva töö tulemustest.

## 5. Tulemused

Käesolevas peatükis antakse ülevaade töö tulemustest. Selgitatakse, kuidas on robot ehitatud ning kuidas see aku tühjenemisel laadimispunkti jõuab.

### 5.1 Roboti ehitamine

Antud punktis antakse ülevaade, kuidas on ühendatud robot erinevad osad. Robot koosneb kahe servomootoriga alusest, kolmandast rattast roboti esiosas, Arduino arendus- ja laiendusplaadist, maketeerimislauast, LCD-tähtekraanist ning sensoritest. Robotil on infrapuna- ning ultraheliandur. Lisaks on roboti ette kinnitatud akupank, mille küljes on juhtmevaba laadimise vastuvõtja.

Arduino arendusplaadile on ühendatud prototüüpimise laiendusplaat koos maketeerimislauaga, et saaks rohkem juhtmeid ühendada. Näiteks on arendusplaadil ainult üks 5V pesa. Maketeerimislaua abil saab ühest pesast signaali edasi kanda mitmesse pesasse.

Arduino arendusplaat on kinnitatud kahepoolse kleeplindiga alumiiniumist alusele. Alus on eelnevalt kaetud isoleerteibiga. Alusel on kaks ratast, mida liigutavad servomootorid. Rattaid on robotil vaja, et liikuda laadimispunkti. Alusele on kinnitatud ka kolmas ratas, et robot oleks tasakaalus.

LCD-tähtekraani ühendamiseks on kasutatud I2C-kontrollerit. Ilma kontrollerita tuleks ühendada rohkem juhtmeid ning väikesel maketeerimislaua ei ole selleks piisavalt vabu pesasid. Ekraan on kinnitatud roboti tagumisse osasse. LCD-tähtekraani peale väljastatakse mõõdetud aku pinge.

Robotil on kaks andurit. Infrapuna sensor on vajalik, et võtta vastu infrapuna signaali. Laadimispunktis on teine Arduino, millele on ühendatud infrapuna saatja. Saatja saadab iga 100 millisekundi tagant kindla väärtusega signaali. Mõlemad infrapuna sensorid on ümbritsetud toruga, et signaali levikut piirata. Vastuvõtja on asetatud roboti ette. Kui vastuvõtja saab signaali, on teada, et laadimispunkt asub roboti ees.

Laadimispunkti otsides võib robot sõita vastu seina. Selle vältimiseks on robotile ühendatud ultraheliandur, mis mõõdab, kui kaugel on roboti ees olev objekt. Ultraheli sensori abil

muudab robot liikumise suunda, kui takistus tuleb ette. Sensor ei mõõda alla 5-sentimeerise laiusega objekte, näiteks toolijalgu või juhtmeid. Seetõttu tuleb sellised esemed roboti liikumisalast enne eemaldada.

Roboti ette on kinnitatud akupank, mille külge on fikseeritud juhtmevaba laadimise vastuvõtja. Laadimispunktis on laadija asetatud risti maaga. Aku tühjenemisel sõidab robot akupangal oleva vastuvõtjaga vastu laadijat ning hakkab laadima.

Joonisel 39 on näha lõplik robot koos anduritega.



**Joonis 39.** Lõplik robot

Käesolevas punktis seletati, kuidas on robot ehitatud. Järgmises punktis kirjeldatakse, kuidas robot laadimispunkti jõuab.

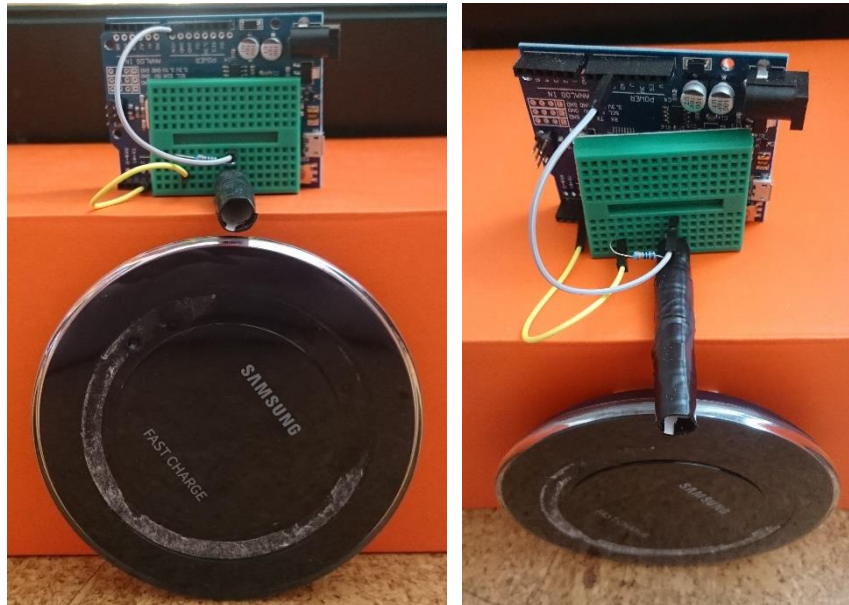
## 5.2 Roboti laadimispunkti jõudmine

Antud punktis antakse ülevaade, kuidas robot laadimispunkti jõuab. Selleks tuleb esmalt tuvastada, et aku tühjeneb ning seejärel leida laadimispunkti asukoht.

Aku pinget tuvastatakse Arduino arendusplaadi abil. Selleks ühendatakse aku positiivne klemm plaadi analoogsisendisse. Analoogsisend tagastab väärtuse 0-1023, mis arvutatakse ümber 0-5 voldiks.

Täislaetud aku pinget on 5 volti. Roboti vabalt liikumisel aku tühjeneb ning väheneb aku pinget. Katseliselt on teada saadud, et kui aku pinget on vähem kui 3.5 volti, robot väga edasi ei liigu. Sel hetkel põleb LCD-tähtekraani taustavalgus tuhmilt ning robot liigub edasi väga aeglaselt. Aku pinget langeb üsna aeglaselt. Seega on piisav, kui robot läheb laadima, kui aku pinget on 3.6 volti.

Laadimispunktis olev Arduino on asetatud juhtmevaba laadija kohale (vt joonis 40). Infrapuna saatja ümber olev toru on suunatud risti laadijaga, et robot liiguks õigesti laadimispunkti.



**Joonis 40.** Laadimispunktis olev Arduino ja juhtmevaba laadija

Laadimispunkti jõudmiseks on robotil kindel algoritm. Kui aku on piisavalt täis, liigub robot vabalt ringi ning väldib takistuste vastu sõitmist. Aku tühjenedes hakkab robot otsima laadimispunkti. Laadimispunkti liikumine on sarnane vabalt liikumisega, kuid iga kord, kui robot liigub otse, hakkab robot koha peal pöörama ning otsib laadimispunktist tulevat signaali. Kui signaal on leitud, liigub robot edasi. Signaali kadumisel korratakse otsimise protsessi.

Robot jõuab küll laadimispunkti, kuid iga kord laadima ei hakka, kuna juhtmevaba laadimise vastuvõtja ja laadija pole täpselt joondatud. See on põhjusel, et praegune tehnoloogia nõuab küllaltki täpset joondust. Tehnoloogia arenedes võib see probleem laheneda. Peale selle saab teha algoritmis muudatusi, et robot liiguks täpsemalt vastu laadijat. Näiteks saab robotile lisada sensori, mille abil liikuda mööda maas olevat joont, et täpsemalt laadimispunkti jõuda.

Käesolevas peatükis anti ülevaade töö tulemustest. Kirjeldati, kuidas roboti erinevad osad on ühendatud ning kuidas robot jõuab aku tühjenedes laadimispunkti.



## Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli koostada juhend, kuidas ehitada robot, mis liigub aku tühjenedes laadimispunkti, kus hakata juhtmevabalt laadima. Kirjeldati Arduino mikrokontrollerplatvormi, mis juhib robotit. Seletati, kuidas Arduinol aku laetust ning tühjenemist tuvastada. Seejärel anti ülevaade juhtmevaba laadimise tehnoloogiatest ning Qi juhtmevaba laadimise standardist. Kirjeldati, kuidas ühendada robot servomootorite ja vajalike anduritega, et laadimispunkti liikuda ning kuidas aku tühjenemisel laadimispunkti leida. Töö lõpus anti ülevaade töö tulemustest.

Töö käigus valmis robot, mis liigub aku tühjenedes laadimispunkti. Selleks tuvastab robot aku laetuse taseme ning kui see on piisavalt tühi, hakkab robot otsima laadimispunkti. Hetkel on veel probleeme töökindlusega ning seetõttu robot vahepeal laadima ei hakka, kuna ei ole täpselt laadijaga joondatud. Käesolev töö on esimene katsetus Tartu Ülikooli arvuteaduse instituudis ehitada juhtmevabalt laadiv robot. Töös keskenduti eelkõige juhtmevaba laadimise tehnoloogia katsetamisele ning seega ei ole antud lahendus kõige esinduslikuma väljanägemisega.

Autori jaoks oli kõige väljakutsuvam osa roboti laadimispunkti liikumine. Seda seetõttu, et robot pidi olema väga täpselt laadijaga joondatud. Peale selle ei andud sensorid alati õigeid tulemusi ning sellega tuli programmi kirjutades arvestada. Siiski oli valitud teema autori jaoks väga põnev ning hariv.

## Kasutatud kirjandus

- [1] M. Metshein, "05 - Arduino - Mis on Arduino?"  
<http://metshein.com/index.php/arvuti/arduino/995-05-arduino-mis-on-arduino>  
(13.02.2016)
- [2] Arduino, "What is Arduino?" <https://www.arduino.cc/en/Guide/Introduction>  
(13.02.2016)
- [3] Arduino, "Frequently Asked Questions" <https://www.arduino.cc/en/Main/FAQ>  
(13.02.2016)
- [4] Arduino, "Language Reference" <https://www.arduino.cc/en/Reference/HomePage>  
(13.02.2016)
- [5] Arduino, "setup()" <https://www.arduino.cc/en/Reference/Setup> (13.02.2016)
- [6] Arduino, "loop()" <https://www.arduino.cc/en/Reference/Loop> (13.02.2016)
- [7] Arduino, "Genuino" <https://www.arduino.cc/en/Main/GenuinoBrand> (16.04.2016)
- [8] M. Banzi, "Massimo Banzi: Fighting for Arduino"  
<http://makezine.com/2015/03/19/massimo-banzi-fighting-for-arduino/> (10.05.2016)
- [9] Z. Romano, "The State of Arduino: A New Sister Brand Announced"  
<https://blog.arduino.cc/2015/05/22/the-state-of-arduino-a-new-sister-brand-announced/>  
(10.05.2016)
- [10] M. Banzi, "Send in The Clones" <https://blog.arduino.cc/2013/07/10/send-in-the-clones/>  
(21.04.2016)
- [11] Sparkfun, "Arduino UNO - R3" <https://www.sparkfun.com/products/11021>  
(22.04.2016)
- [12] Amazon, "DCCduino ATMEGA328 Development Board For Arduino UNO R3"

- <http://www.amazon.com/dp/B010B2PSYE> (11.03.2016)
- [13] Amazon, "Arduino UNO R3 Board Module With DIP ATmega328P (Blue)"  
<http://amzn.to/1cDU75T> (08.03.2016)
- [14] Arduino, "Arduino Products" <https://www.arduino.cc/en/Main/Products> (13.02.2016)
- [15] Arduino, "Compare board specs" <https://www.arduino.cc/en/Products/Compare>  
(22.04.2016)
- [16] Sparkfun, "Arduino 101" <https://www.sparkfun.com/products/13787> (22.04.2016)
- [17] Sparkfun, "Arduino Pro 328 - 5V/16MHz" <https://www.sparkfun.com/products/10915>  
(22.04.2016)
- [18] Arduino, "Genuino Mega 2560 Rev 3" <https://store.arduino.cc/product/GBX00067>  
(22.04.2016)
- [19] Arduino, "Genuino Zero" <https://store.arduino.cc/product/GBX00003> (22.04.2016)
- [20] Arduino, "Arduino UNO" <https://www.arduino.cc/en/Main/ArduinoBoardUno>  
(22.04.2016)
- [21] Arduino, "Arduino Starter Kit" <https://www.arduino.cc/en/Main/ArduinoStarterKit>  
(22.04.2016)
- [22] Arduino, "Arduino 101" <https://www.arduino.cc/en/Main/ArduinoBoard101>  
(22.04.2016)
- [23] Arduino, "Arduino Pro" <https://www.arduino.cc/en/Main/ArduinoBoardPro>  
(22.04.2016)
- [24] Arduino, "Arduino Mega 2560"  
<https://www.arduino.cc/en/Main/ArduinoBoardMega2560> (22.04.2016)
- [25] Arduino, "Arduino Zero" <https://www.arduino.cc/en/Main/ArduinoBoardZero>

(22.04.2016)

- [26] Arduino, "Shields" <https://www.arduino.cc/en/Main/ArduinoShields> (28.02.2016)
- [27] Arduino, "Hardware & Related Initiatives"  
<http://playground.arduino.cc/Main/SimilarBoards#goShie> (28.02.2016)
- [28] Arduino, "Arduino Proto Shield" <https://www.arduino.cc/en/Main/ArduinoProtoShield>  
(28.02.2016)
- [29] Amazon, "uxcell® Prototyping Prototype Shield ProtoShield w Mini Breadboard"  
[http://www.amazon.com/dp/B00SWNZXZO/ref=cm\\_sw\\_su\\_dp](http://www.amazon.com/dp/B00SWNZXZO/ref=cm_sw_su_dp) (11.03.2016)
- [30] Codeduino, "Arduino vs Raspberry Pi Comparison" <https://codeduino.com/information-and-news/hardware/arduino-vs-raspberry-pi/> (08.03.2016)
- [31] Arduino, "Getting Started with Arduino and Genuino on Windows"  
<https://www.arduino.cc/en/Guide/Windows> (08.03.2016)
- [32] M. Rundle, "Raspberry 3 is the first with built-in Wi-Fi"  
<http://www.wired.co.uk/news/archive/2016-02/29/raspberry-pi-three-wifi-bluetooth-release-price-cost> (22.04.2016)
- [33] Sparkfun, "Analog to Digital Conversion" <https://learn.sparkfun.com/tutorials/analog-to-digital-conversion> (22.04.2016)
- [34] LinuxUser, "Python explained" <http://www.linuxuser.co.uk/raspberry-pi/python-explained> (22.04.2016)
- [35] The MagPi Magazine, "Raspberry Pi 3 is out now! Specs, benchmarks & more"  
<https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/> (22.04.2016)
- [36] L. Orsini, "Arduino Vs. Raspberry Pi: Which Is The Right DIY Platform For You?"  
<http://readwrite.com/2014/05/07/arduino-vs-raspberry-pi-projects-diy-platform/>  
(22.04.2016)

- [37] Raspberry Pi, "GPIO: Models A+, B+, Raspberry Pi 2 B and Raspberry Pi 3 B"  
<https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/> (10.05.2016)
- [38] Amazon, "Raspberry Pi 3 Model B Mother board" <http://amzn.com/B01CD5VC92>  
(22.04.2016)
- [39] The Human Touch of Chemistry, "How do rechargeable batteries work?"  
<http://humantouchofchemistry.com/how-do-rechargeable-batteries-work.htm>  
(06.03.2016)
- [40] UCDAVIS Chem Wiki, "Rechargeable batteries"  
[http://chemwiki.ucdavis.edu/Core/Analytical\\_Chemistry/Electrochemistry/Voltaic\\_Cell/s/Case\\_Study%3A\\_Battery\\_Types/Rechargeable\\_Batteries](http://chemwiki.ucdavis.edu/Core/Analytical_Chemistry/Electrochemistry/Voltaic_Cell/s/Case_Study%3A_Battery_Types/Rechargeable_Batteries) (06.03.2016)
- [41] Eesti entsüklopeedia, "elektrood" <http://entsyklopeedia.ee/artikkel/elektrood2>  
(06.03.2016)
- [42] Power System, "How batteries work?"  
<http://www.qrg.northwestern.edu/projects/vss/docs/power/2-how-do-batteries-work.html> (06.03.2016)
- [43] Network of Excellence Robotic & Mehatronics Homelab Community, "Akude laadimisest" [http://home.roboticlab.eu/media/et/abimaterjalid/akude\\_laadimine.pdf](http://home.roboticlab.eu/media/et/abimaterjalid/akude_laadimine.pdf)  
(14.02.2016)
- [44] J. Heald, "Voltage Sources: Energy Conversion and Examples"  
<http://study.com/academy/lesson/voltage-sources-energy-conversion-and-examples.html> (14.02.2016)
- [45] M. Metshein, "11 - Arduino - Projekt 5: sumisti ja potentsiomeeter"  
<http://metshein.com/index.php/arvuti/arduino/1001-11-arduino-projekt-5-sumisti-ja-potensiomeeter> (02.04.2016)
- [46] Android, "analogRead()" <https://www.arduino.cc/en/Reference/AnalogRead>

(10.05.2016)

- [47] Arduino, "Read Analog Voltage"  
<https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage> (08.05.2016)
- [48] Sparkfun, "I2C" <https://learn.sparkfun.com/tutorials/i2c> (03.05.2016)
- [49] Arduino-info Wiki, "LCD-Blue-I2C" <https://arduino-info.wikispaces.com/LCD-Blue-I2C> (03.04.2016)
- [50] Arduino e Cia, "Como utilizar o módulo I2C com display LCD"  
<http://www.arduinoecia.com.br/2014/12/modulo-i2c-display-16x2-arduino.html>  
(03.04.2016)
- [51] BitBucket, "New LiquidCrystal" <https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads> (03.05.2016)
- [52] C. Simpson, "Characteristics of Rechargeable Batteries"  
<http://www.ti.com/lit/an/snva533/snva533.pdf> (22.04.2016)
- [53] Battery University, "BU-501: Basics about Discharging"  
[http://batteryuniversity.com/learn/article/discharge\\_methods](http://batteryuniversity.com/learn/article/discharge_methods) (22.04.2016)
- [54] L. Xie, Y. Shi, Y. T. Hou, W. Lou, "Wireless power transfer and applications to sensor networks," *Wireless Communications, IEEE*, vol. 20, no. 4, pp. 140-145, 2013.
- [55] J. Garcia, R. A. Chinga, J. Lin, "Wireless Power Transmission: From Far Field to Near Field," *Proceedings of the IEEE*, vol. 101, no. 6, pp. 1321-1331, 2013.
- [56] Tesla Memorial Society of New York, "Wardenclyffe - A Forfeited Dream"  
<http://www.teslasociety.com/wardenclyffe2.htm> (28.02.2016)
- [57] A. Tomar and S. Gupta, "Wireless power Transmission: Applications and Components," *International Journal of Engineering Research and Technology, ESRSA*, vol. 1, no. 5, 2012.

- [58] N. M. Tahsin, M. M. Siddiqui, M. A. Zaman, M. I. Kayes, "Wireless Charger for Low Power Devices using Inductive Coupling"  
[http://www.academia.edu/2329757/Wireless\\_Charger\\_for\\_low\\_power\\_devices\\_using\\_inductive\\_coupling](http://www.academia.edu/2329757/Wireless_Charger_for_low_power_devices_using_inductive_coupling) (12.05.2016)
- [59] C. Woodford, "Wireless induction chargers"  
<http://www.explainthatstuff.com/inductionchargers.html> (27.04.2016)
- [60] X. Lu, D. Niyato, P. Wang, D. I. Kim, Z. Han, "Wireless charger networking for mobile devices: fundamentals, standards, and applications," *Wireless Communications, IEEE*, vol. 22, no. 2, pp. 126-135, 2015.
- [61] Magnet Academy, "Electromagnet induction"  
<https://nationalmaglab.org/education/magnet-academy/watch-play/interactive/electromagnetic-induction> (29.11.2015)
- [62] Online Electrical Engineering, "Faraday Law of Electromagnetic Induction"  
<http://www.electrical4u.com/faraday-law-of-electromagnetic-induction/> (12.12.2015)
- [63] The Columbia Encyclopedia, 6th ed., "Galvanometer"  
<http://www.encyclopedia.com/topic/galvanometer.aspx#1> (12.12.2015)
- [64] HyperPhysics, "Faraday's law" <http://hyperphysics.phy-astr.gsu.edu/hbase/electric/farlaw.html> (12.12.2015)
- [65] Wireless Power Consortium, "The Benefits Of Qi"  
<http://www.wirelesspowerconsortium.com/about/benefits.html> (11.01.2016)
- [66] Wireless Power Consortium, "A Qi wireless charger: Resonant as well as Inductive"  
<http://www.wirelesspowerconsortium.com/technology/qi-wireless-charger-resonant-as-well-as-inductive.html> (11.01.2016)
- [67] Wireless Power Consortium, "Wireless Electricity Transmission"  
<http://www.wirelesspowerconsortium.com/technology/how-it-works.html> (15.11.2015)

- [68] Amazon, "Samsung EP-PG920IBUGUS Wireless Charging Pad with 2A Wall Charger - Black Sapphire" <http://www.amazon.com/Samsung-EP-PG920IBUGUS-Wireless-Charging-Charger/dp/B00UCZGS6S> (30.04.2016)
- [69] Lazada, "Universal Qi Wireless Charging Receiver Kit for Micro-USB Android Mobiles Down (Black)" <http://www.lazada.com.my/universal-qi-wireless-charging-receiver-kit-for-micro-usb-androidmobiles-down-black-2819959.html> (30.04.2016)
- [70] Arduino, "Sweep" <https://www.arduino.cc/en/Tutorial/Sweep> (06.03.2016)
- [71] Robotech Shop, "Mini Breadboard" <http://robotechshop.com/shop/%E2%80%A2prototyping/boards/mini-breadboard/> (25.04.2016)
- [72] M. Metshein, "15 - Arduino - Projekt 10: servomootorid" <http://metshein.com/index.php/arvuti/arduino/1005-15-arduino-projekt-10-servomootorid> (26.04.2016)
- [73] SB-Projects, "IR Remote Control Theory" <http://www.sbprojects.com/knowledge/ir/index.php> (25.04.2016)
- [74] Sumeet, "Infrared Transmitter & Receiver Pair" [http://www.sumeetinstruments.com/shop/index.php?route=product/product&product\\_id=166](http://www.sumeetinstruments.com/shop/index.php?route=product/product&product_id=166) (25.04.2016)
- [75] Arduino-info Wiki, "Infrared (IR) Remote" <https://arduino-info.wikispaces.com/IR-RemoteControl> (25.04.2016)
- [76] Adafruit, "Using an IR Sensor" <https://learn.adafruit.com/ir-sensor/using-an-ir-sensor> (25.04.2016)
- [77] Sparkfun, "IR Communication" <https://learn.sparkfun.com/tutorials/ir-communication> (25.04.2016)
- [78] Openlgtv, "LG TV USB IR-Hack with Arduino"



[http://openlgtv.org.ru/wiki/index.php/LG\\_TV\\_USB\\_IR-Hack\\_with\\_Arduino](http://openlgtv.org.ru/wiki/index.php/LG_TV_USB_IR-Hack_with_Arduino)

(25.04.2016)

[79] Github, "Arduino-IRremote" <https://github.com/shirriff/Arduino-IRremote> (03.05.2016)

[80] Ebay, "IR Infrared Wireless Power Remote Control Kits 38KHz Sensor Board for Arduino AVR PIC" <http://www.ebay.ie/itm/IR-Infrared-Wireless-Remote-Control-Kits-38KHZ-Sensor-Board-for-Arduino-AVR-PIC-/121673399764> (29.04.2016)

[81] M. Metshein, "27 - Arduino - Projekt 22: ultraheli- ja infrapunaandur" <http://metshein.com/index.php/arvuti/arduino/1017-27-arduino-projekt-22-ultraheli-ja-infrapunaandur> (29.04.2016)

[82] RoboLink, "HC-SR04 Ultrasonik Mesafe Senörü" <https://www.robolinkmarket.com/arduino-ultrasonic-mesafe-sensoru-hc-sr04.html> (29.04.2016)

# Lisad

## I. Roboti kood

Programmi kood on kättesaadav ka Tartu Ülikooli arvutiteaduse instituudi lõputööderegistris.

```
/* Juhtmevaba iselaadiv robot
 *
 * Programmi abil tuvastab robot aku tühjenemise ning
 * läheb laadimispunkti juhtmevabalt laadima
 *
 * Autor: Triinu Liis Kelder
 * Programmis on kasutatud näidisprogramme, mis on avalikult kättesaadavad
https://www.arduino.cc/en/Tutorial/BuiltInExamples
 * ning http://metshein.com/index.php/arvuti/arduino
 *
 * 2016
 */

#include <IRremote.h>
#include <IRremoteInt.h>
// https://arduino-info.wikispaces.com/IR-RemoteControl
// RobotIRremote tuleb ära võtta (C:\Program Files
// (x86)\Arduino\Libraries\RobotIRremote), et saaks seda teeki kasutada

#include <Servo.h> // Arduino IDE-s olemas

#include <Wire.h> // Arduino IDE-s olemas
#include <LiquidCrystal_I2C.h> // Tuleb eraldi installida
// (https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads)

// LCD ekraan
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Ühendab LCD-
ekraani õigete pesadega ära

// Servo
Servo servoLeft; // Loob servo objekti
Servo servoRight; // Loob servo objekti

// Aku pinget leidmine
const int analogInPin = A0; // Määrab ära, millisest pesast pinget väärtus
loetakse
int sensorValue = 0; // Loob sensorValue muutuja

// IR vastuvõtja
int RECV_PIN = 11; // Määrab ära sensori pesa
IRrecv irrecv(RECV_PIN); // Teeb irrecv objekti
decode_results results; // Teeb results objekti

int no_signal = 0;
```

```

// Ultraheli
int EchoPin = 12; // Määrab õiged pesad
int TrigPin = 13;
long EchoTime = 0;

bool otsi = false; // Muutuja, kas tuleb otsida laadimispunkti või ei

bool paremale = true; // Muutuja, kas tuleb pöörata paremale või mitte
int turn_counter = 0; // Muutuja, palju on robot pööranud
bool leitud = false; // Muutuja, kas laadimispunkti signaal on leitud või mitte
bool kohal = false; // Muutuja, kas robot on jõudnud laadimispunkti või mitte

int takistus = false; // Muutuja, kas takistus on lähedal või ei

void setup() {
  Serial.begin(9600); // Initsialiseerib jadapordi monitori
  servoLeft.attach(7); // Ühendab servo objekti pesaga 7
  servoRight.attach(9); // Ühendab servo objekti pesaga 9
  lcd.begin(16,2); // Initsialiseerib ekraani 16 veeru ja 2 reaga
  irrecv.enableIRIn(); // Alustab IR vastuvõtja sensori

  // Ultrasonic
  pinMode(EchoPin, INPUT);
  pinMode(TrigPin, OUTPUT);
  stopRobot();
}

void loop() {
  lcd.setCursor(0, 0);
  lcd.backlight();
  readVoltage();
  for (int i = 0; i < 40; i++) {
    is_signal();
    delay(50);
  }
}

// Väljastab true kui robot on liiga lähedal takistusele ning false kui ei ole
// veel liiga lähedal
int ultrasonic() {
  digitalWrite(TrigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(TrigPin, LOW);

  EchoTime = pulseIn(EchoPin, HIGH);
  int dist = (EchoTime/2)/29;
  Serial.print(dist, DEC);
  Serial.println("cm");
  return dist;
}

void is_signal() {
  if (irrecv.decode(&results)) {
    switch(results.value) {

```

```

case 0xa90: // Signaal, mis tuleb infrapuna saatjast
    // Kui tuleb otsida, kontrollib, kas takistus on lähedal
    // Kui takistus on kaugel, liigub edasi 50 ms
    // Kui takistus lähedal, peatab roboti
    if (otsi) {
        leitud = true;
        // Kontrollib, ega robot liiga lähedal laadimispunktile pole
        if (ultrasonic() > 15 || ultrasonic() == 0) {
            forward(50);
            // Robot on peaaegu laadimispunktis
            // Edasi liigub ainult otse, kuni jõuab laadijani
        } else {
            while (ultrasonic() > 2) {
                forward(50); // Robot läheb edasi kuni jõuab laadimispunktist 2 cm
            }
            forward(20); // Natukene veel, et oleks vastu laadijat
            kohal = true; // Laadimispunkt on leitud
            otsi = false; // Otsida pole enam vaja
            stopRobot();
        }
        no_signal = 0;
    }
    break;
case 0xFD8877: // Üles nool puldil
    forward(200);
    stopRobot();
    break;
case 0xFD6897: // Paremale nool puldil
    turnRight(400);
    stopRobot();
    break;
case 0xFD28D7: // Vasakule nool puldil
    turnLeft(400);
    stopRobot();
    break;
case 0xFD9867: // Alla nool puldil
    reverse(200);
    stopRobot();
    break;
case 0xFDA857: // OK nupp puldil
    stopRobot();
    break;
case 0xFD30CF: // * nupp puldil
    otsi = true;
    break;
case 0xFD708F: // # nupp puldil
    otsi = false;
    break;
default: // Mingi teine väärtus leiti
    // Kui tuleb otsida, peatab roboti ja suurendab no_signal väärtust
    if (otsi) {
        stopRobot();
        no_signal++;
    }

```

```

    }
    Serial.println(results.value, HEX);
}
} else { // Signaali ei leitud
    // Tuleb otsida, aga signaali ei leitud.
    // Peatab roboti ja suurendab no_signal väärtust
    if (otsi) {
        stopRobot();
        no_signal++;
    }
}
irrecv.resume(); // Otsib uuesti signaali

// Robot liigub
if (!kohal) {
    liigu();
}
}

void liigu() {
    takistus = ultrasonic();
    if (takistus > 10 || takistus == 0 || leitud) { // Kas takistus on lähedal
        // Kui signaali pole leitud 10 korda järjest ja tuleb otsida ja pole täisringi
        // veel teinud
        if (no_signal > 9 && (otsi || leitud) && turn_counter < 120) {
            turnRight(40);
            turn_counter++;
            stopRobot();
        } else {
            forward(50);
            turn_counter = 0;
        }
    } else {
        stopRobot();
        if (paremale) { // Tuleb pöörata paremale
            turnRight(400); // Pöörab paremale
            stopRobot();
            takistus = ultrasonic();
            if (takistus > 10 || takistus == 0) { // Kontrollib, kas on takistus
                lähedal
                forward(50);
                turnRight(400);
                stopRobot();
                paremale = false;
            } else { // Kui takistus tuli liiga lähedale, proovib vasakule pöörata
                turnLeft(800);
                stopRobot();
                takistus = ultrasonic();
                if (takistus > 10 || takistus == 0) { // Kontrollib, kas on takistus
                    lähedal
                    turnLeft(350);
                    stopRobot();
                } else { // Kui takistus tuli liiga lähedale, pöörab tagasi otse, liigub
                    tagasi

```

```

        turnRight(400);
        stopRobot();
        reverse(50);
    }
}
} else { // Tuleb pöörata vasakule
    turnLeft(400);
    stopRobot();
    takistus = ultrasonic();
    if (takistus > 10 || takistus == 0) { // Kontrollib, kas on takistus
Lähedal
        forward(50);
        turnLeft(400);
        stopRobot();
        paremale = true;
    } else { // Kui takistus tuli liiga lähedale, proovib paremale pöörata
        turnRight(700);
        stopRobot();
        takistus = ultrasonic();
        if (takistus > 10 || takistus == 0) { // Kontrollib, kas on takistus
Lähedal
            turnRight(400);
            stopRobot();
        } else { // Kui takistus tuli liiga lähedale, pöörab tagasi otse, liigub
tagasi
            turnLeft(400);
            stopRobot();
            reverse(50);
        }
    }
}
}
}
}

// Võtab argumendiks rea numbri ning
// kustutab sellel real oleva sõne
void emptyAndSetRow(int row) {
    lcd.setCursor(0, row);
    lcd.print(" ");
    lcd.setCursor(0, row);
}

// Loeb aku pinget ning väljastab selle LCD-ekraanile
void readVoltage() {
    sensorValue = analogRead(analogInPin);
    // Kui aku pinge on alla 3.6 voldi, hakkab robot otsima laadimispunkti
    if ((sensorValue / 1023.0 * 5.0) < 3.6) {
        otsi = true;
    }
    // Väljastab LCD-ekraanile aku pinge völdides
    emptyAndSetRow(0);
    lcd.print(sensorValue / 1024.0 * 5.0);
    lcd.print("V");
}

```

```

// Võtab argumendiks täisarvu ning
// sõidab nii kaua edasi (millisekundites)
void forward(int n) {
    servoLeft.write(180);
    servoRight.write(7);
    delay(n);
}

// Võtab argumendiks täisarvu ning
// sõidab nii kaua tagasi (millisekundites)
void reverse(int n) {
    servoLeft.write(7);
    servoRight.write(180);
    delay(n);
}

// Võtab argumendiks täisarvu ning
// pöörab nii kaua paremale (millisekundites)
void turnRight(int n) {
    servoLeft.write(180);
    servoRight.write(180);
    delay(n);
}

// Võtab argumendiks täisarvu ning
// pöörab nii kaua vasakule (millisekundites)
void turnLeft(int n) {
    servoLeft.write(0);
    servoRight.write(0);
    delay(n);
}

// Peatab roboti
void stopRobot() {
    servoLeft.write(93);
    servoRight.write(93);
}

```

## **II. Litsents**

### **Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, **Triinu Liis Kelder** (sünnikuupäev 23.07.1994)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose Juhtmevaba iselaadiva Arduino roboti ehitamine, mille juhendajad on Anne Villem, Taavi Duvin ja Alo Peets,
  - 1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **12.05.2016**